PC98シリーズ対応版

とことんこだわる
CONFIG.SYS

NEC版MS-DOS6.2対応

寺口俊伸著



■HSB：高速リブート ■HSBM：HSBメニュー
■VMAP：メモリ情報の表示
■SETPATH：長いパスを設定
■ENV_DRV：ドライブ情報を環境変数に登録
■オリジナルCONFIG.SYS多数収録

3.5インチディスク付
2HD1.25MB MS-DOSフォーマット

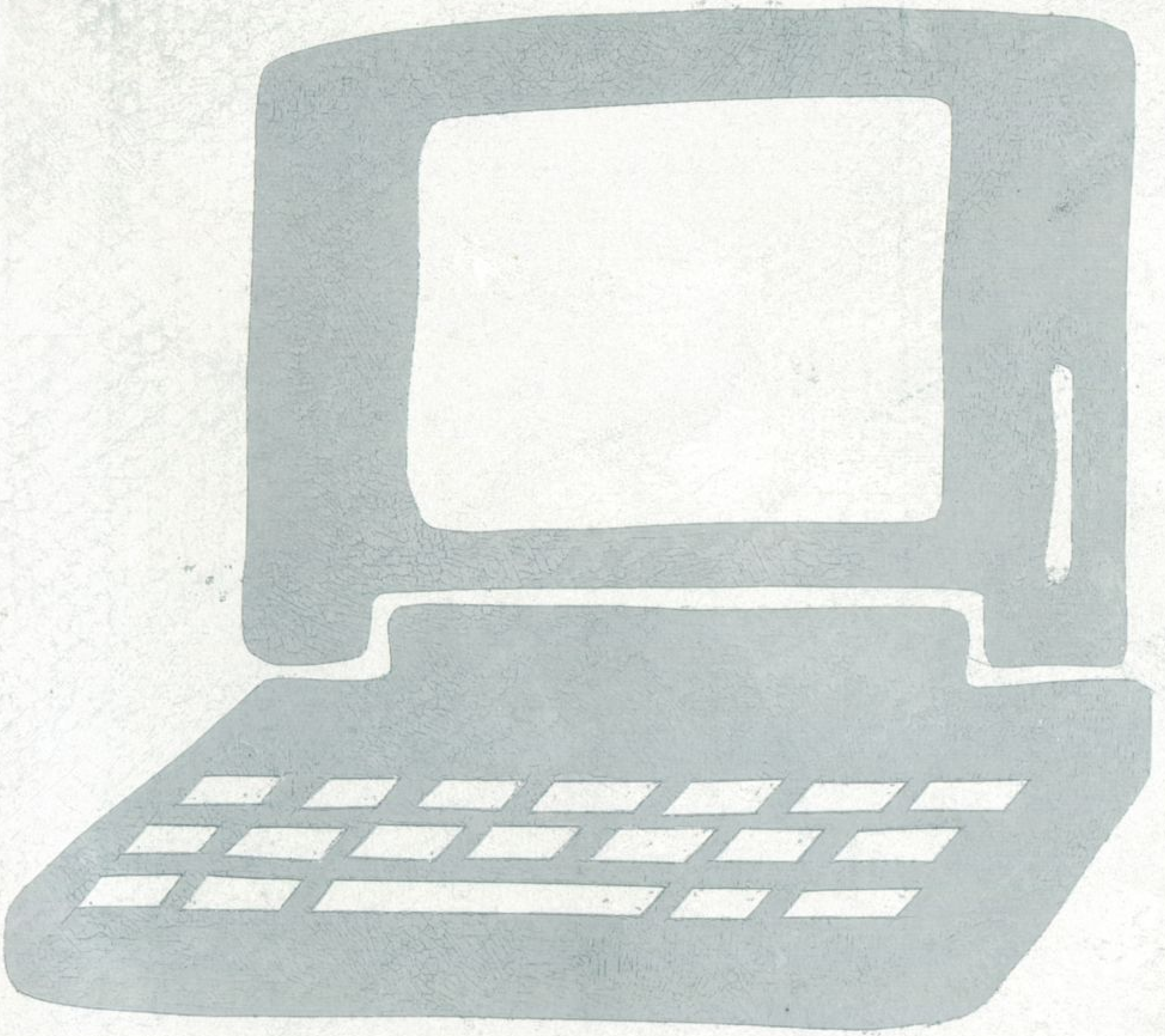
PC98
シリーズ
対応版

ここまでこだわる

CONFIG.SYS

NEC版MS-DOS6.2対応

寺口俊伸著



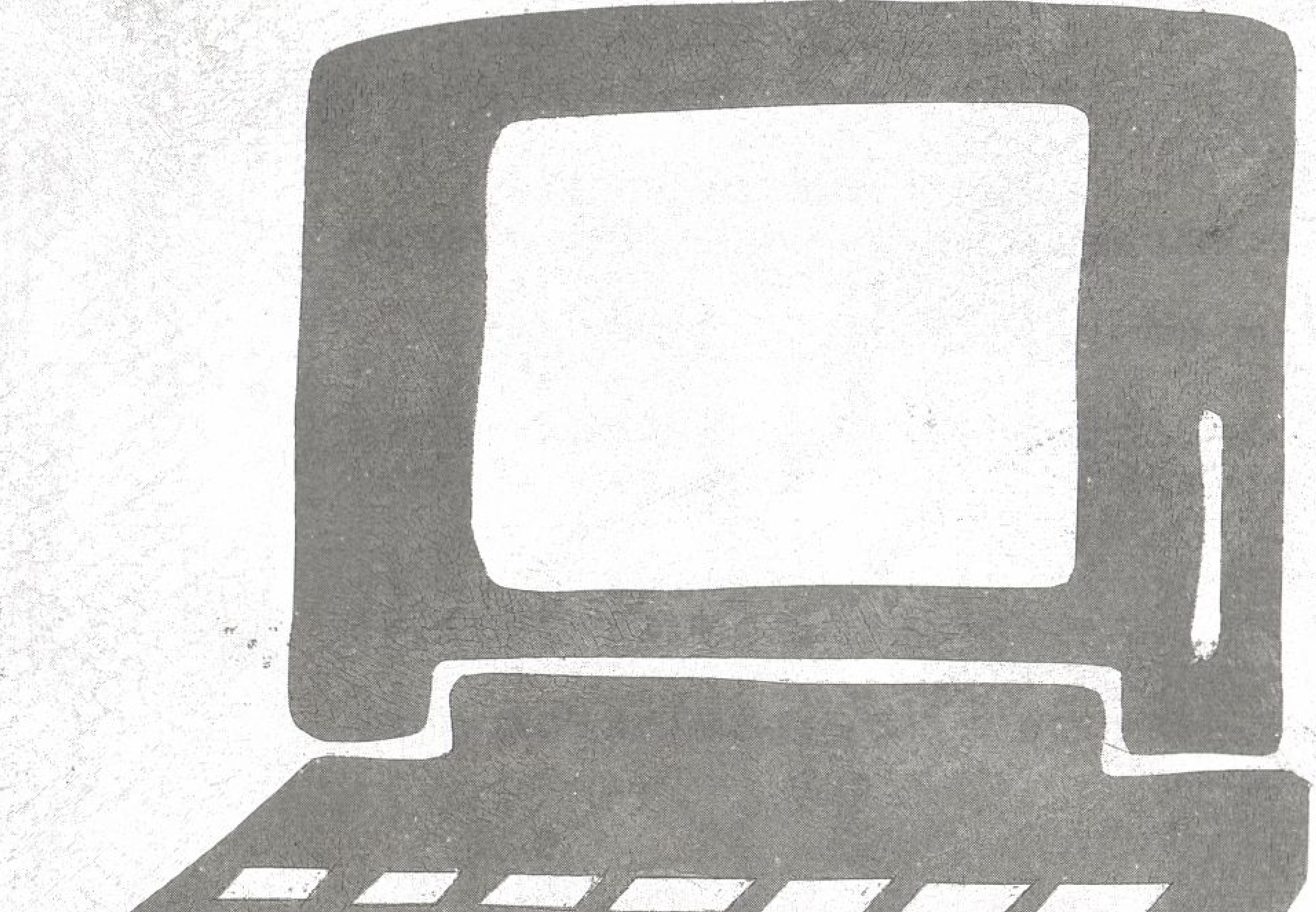
PC98シリーズ対応版

とことんこだわる

CONFIG.SYS

NEC版MS-DOS6.2対応

寺口俊伸著



◉本書で使用したハードとソフト

パソコン本体：PC-9821Xa/C10W、PC-9821Ap/U2、PC-9821Lt/540A(NEC)
MO　　　　　：LF-3200JD（Panasonic）
PD　　　　　：LF-1000JD（Panasonic）
SCSIカード　：PSC-SCSI/P（Caravelle）
モデム　　　：MC288XL（MICRO CORE）
プリンタ　　：MEDIO GP55F（Canon）
OS　　　　　：MS-DOS Ver.6.2、Windows Ver.3.1（NEC）

◉権利について

本書で取り上げた主なソフトウェアおよびハードウェアの商標または登録商標、ならびに商標権者、開発者は以下のとおりです。

MS-DOS、Windows：米国 Microsoft Corporation
Intel 8086/80286/80386/80486/Pentium：米国 Intel Corporation
VJE-Delta：株式会社バックス
ATOK8：株式会社ジャストシステム
WX III：エー・アイ・ソフト株式会社
MEMORY SERVER II：株式会社アイ・オー・データ機器
MELWARE for Windows：株式会社メルコ
SETPATH：Kenny（BOSTON倶楽部)/フリーソフトウェア
ENV_DRV：山下良輔（ryama)/フリーソフトウェア
HSB：森河正男（Masao)/フリーソフトウェア
HSBM：KEi/フリーソフトウェア
VMAP：兵藤嘉彦（c.mos)/フリーソフトウェア

その他、本文中で使用の製品および製品名は、一般に各社の商標および登録商標です。
なお、本文中では™、Ⓡマークは明記していません。

◉内容について

本書の内容につきましては出版元であるエーアイ出版編集部と著者が上記の環境で実際に操作した結果を慎重に検討し、著述・編集したものです。本書の内容に関して、誤り・質問などがありましたら出版元まで必ずFAXにてご連絡ください。ただし、ソフトウェアやハードウェアそのものの機能や本書に記述されていない操作については、それぞれのソフトウェア・ハードウェアに付属の説明書をご覧になるか、直接メーカーにお問い合わせください。

また、本書内容の運用結果につきましては、いかなる場合も責任を負いかねますのでご了承ください。

はじめに

本書は、“とことんこだわるシリーズ”の第6作目です。

タイトルからすると、「とことんこだわるCONFIG.SYS DOS5編（改訂増補）」をDOS6対応として改めただけのように思われるかもしれません。しかし、中を見て驚かれた方もいらっしゃることでしょう。内容もスタイルも、DOS5編とは大きく変えてありますから。

たとえば、DOS5編では主にメモリにこだわった解説を行っていましたが、本書はどちらかというと周辺機器などによるシステムの拡張についての解説が中心になっています。

それには、2つの理由があります。ひとつは、DOS6にMemMakerというユーティリティが付属になり、誰でも簡単にメモリの最適化が行えるようになったということ。もうひとつは、最近のシステムは拡張メモリが十分に搭載されていることが多くなり、従来ほどメモリに関して神経を遣う必要がなくなってきたということです。

筆者には、いまだに手作業によるメモリの設定にこだわりたいという気持ちもありますが、こうした流れを無視するするのも愚かなことです。やはり、こだわりの方向を変えるのが妥当ではないかと考えました。

執筆が遅れてしまった分、ギリギリで入手できたPD装置をはじめ、ノートパソコンで使われているPCカード、それから、DOS6対応版が出た『MEMORY SERVER II』など、最新の情報を盛り込むことができています。

また、本書は従来のQ&A形式ではなく、ステップアップ形式を採用することにしました。それには、次のような理由があります。

当初、この“とことんこだわるシリーズ”は、初心者の段階を卒業されたかたがたに対して、実用面から多少離れた、“こだわり”の世界へとお誘いしてみようという意図で始めました。目指したのは、初心者向けの入門書と、上級者向けの技術解説書の中間くらいの存在です。そのためには、“学ぶ”というより“楽しむ”という姿勢で読んでいただくほうがよいだろうと考え、Q&A形式を用いることにしたのです。

ところが、いざ出版してみると、筆者が想定していなかった初心者と思われる方からも多くの読者ハガキが送られてきまして、たいへんありがたく思ったのと同時に、申し訳ない気持ちでいっぱいになりました。どうやら、筆者が考えていた“こだわり”の世界は、多くのユーザーにとって通らねばならない道の先にあったようです。

初心者の方にも、少なくとも途中までは迷わずに読み進めていただけるように、なんとか情報量を増やして、しかも、見ただけで敬遠したくなるような厚さの本にはしたくない……思い悩んだ末に、少しでも効率のよいスタイルとしてステップアップ形式が浮かんだのです。幸いなことに、初めてステップアップ形式を用いた、『とことんこだわるCONFIG.SYS DOS/V編』はたいへんご好評をいただきました（DOS/V機とPC-98の両方をお持ちの方には、両方の本を読み比べていただけると、それぞれの特徴が理解しやすいと思います）。

●

末筆ながら、本書を執筆する機会を与えてくださったエーアイ出版（株）編集部のみなさん、そして、解説とディスクへの収録をお許しくださったフリーソフトウェアの作者のみなさんに、心よりお礼申し上げます。

1995年6月22日

著者しるす

※フリーソフトウェアを使った感想を、ぜひアンケート葉書でお寄せください。

パソコン通信のサポート会議室に書き込んでいただいても結構ですが、作者に直接電子メールを送ることは避けていただくようお願いします。

本書の読み方

本書は、次の4部構成になっています。

● Part 1　基礎編（Step 1～31）

CONFIG.SYSやAUTOEXEC.BATの最も基本的な内容について解説しています。この基礎編にある内容は、すべてマスターするようにがんばってください。

● Part 2　実用編（Step32～41）

システムの構成や利用目的の違いによって、どのような設定を行えばよいのか、具体的な設定例を示して解説しています。みなさんの環境に合わせて、必要な部分だけを参照してください。

● Part 3　応用編（Step42～49）

ここから先、みなさんは“こだわり”の世界へと入ってゆくことになります。まず最初は、迷路のような世界で迷わないように、メモリに関係のあるコンピュータ用語についての解説を行っています。ここをガイドブックと思って、迷いそうになったら読み返すようにしてください。

それから、いよいよMemMakerを使っていただくのですが、こだわり派のあなたは、ぜひともカスタムセットアップにチャレンジしていただきたいと思います。

● Part 4　Other Tools編（Step50～56）

このOther Tools編では、まずMS-DOS Ver.6.2に付属の拡張メモリマネージャの代わりに、『MEMORY　SERVER II』または『MELWARE　for Windows』を使用する場合の設定方法を解説しています。

次に、付属のディスクに収められているフリーソフトウェアについての解説も行っていますが、それらのフリーソフトウェアは、必ずやみなさんの環境改善に役立ってくれることでしょう。ぜひ使ってみてください。

CONTENTS

Part 1 基礎編

Part 2 実用編

Part 3 応用編

Part4 Other Tools 編

Part 1
基礎編

基礎編の内容

この基礎編では、主に従来のDOSとMS-DOS Ver.6.2との違いについて解説します。特に、MS-DOS Ver.3.3からバージョンアップしたユーザーは、ハードディスクの領域に関する注意をよく読んでおいてください。

次に、CONFIG.SYSとAUTOEXEC.BATの基本的なコマンドや、MS-DOS Ver.6.2に付属の各種デバイスドライバおよび常駐プログラムの使い方について解説します。次の実用編では、状況別の具体的な設定へと進みますが、それには、この基礎編の内容を理解していることが肝心です。

Step 1

DOS6の特徴

●ポイント●

- MS-DOS Ver.3.3DとMS-DOS Ver.6.2では、基本的な仕様が大きく異なっている。特に、管理できるハードディスクの容量が128Mバイトから2Gバイト(2048Mバイト)に拡張されている点に注意。
- MS-DOS Ver.5.0AとMS-DOS Ver.6.2では、基本的な仕様はそれほど変わっていないが、便利なユーティリティが加わっている。
- MS-DOS Ver.6.2で作成したハードディスク内で128Mバイトを超えた領域は、MS-DOS Ver.3.3D以前のDOSでは認識されないので注意が必要。

●解 説●

複数バージョンのMS-DOSが販売されている理由

本書で解説するMS-DOS Ver.6.2は、MS-DOS Ver.3.3DやMS-DOS Ver.5.0Aなど従来のバージョンと、どのような違いがあるのでしょうか。それぞれのバージョンからの変更点を挙げてみましょう。場合によっては、特にバージョンアップする必要がないこともありうると思います。

なお、MS-DOS Ver.6.2というのは、ちょっと長すぎますから、本書でのMS-DOSの各バージョンの略称を、次のように決めておくことにします。

- MS-DOS Ver.3.3D → DOS3
- MS-DOS Ver.5.0A → DOS5
- MS-DOS Ver.6.2 → DOS6

単に「**MS-DOS**」とあれば、すべてのバージョンに共通の話と考えてください。

● DOS3は旧機種のためのもの

さて、DOS6が発売される以前、MS-DOSではバージョン3.3D、バージョン5.0Aの2種類が販売されていました。本来なら、新しいバージョンが発売されると、旧バージョンの販売は停止されるはずです。

しかし、DOS3からDOS5へのバージョンアップでは、メモリやディスクに関する仕様が大幅に変更されたため、DOS3対応のソフトウェアのうち、DOS5では動作しないものが発生じてしまったのです。そこで、やむをえず旧バージョンの販売も継続してきたのだと考えられます。DOS6が登場して、店頭でのDOS5との入れ替えは進むでしょうが、おそらく、しばらくの間はDOS3の販売は続けられるのではないかと思います。

ただ、DOS3というのは、主に旧機種の利用者のために販売されているDOSと考えられます。ハードディスクの容量が最大で128Mバイトまでであったり、16Mバイトを超えるメモリに対応できないなど、DOS3では最近のシステムを活用することはできないのです。新しくパソコンを購入したり、買い替えたりした場合には、必ずDOS5以降を利用するほうがよいでしょう。

DOS3からDOS5への大幅な変更に比べ、DOS5からDOS6への変更点はそれほど大きくありません。基本的には、DOS5対応のアプリケーションで、DOS6になって使えなくなるものはごくわずだかと考えられます。ただし、プログラムの起動時にDOSのバージョンをチェックする仕様になっているものは、サポート外のバージョンと判断されて、起動できないこともあります。その場合には、後述するSETVER.EXEを試してください。

DOS6での変更点

DOS6において、DOS3あるいはDOS5のユーザーが特に考慮すべき、仕様の変更点は次のとおりです。

● DOS3との相違

- ハードディスクの使用可能容量が最大128Mバイトから2Gバイト(2048Mバイト)に拡張された。
- 拡張メモリ仕様のひとつであるXMSに対応した。
- DOSのシステムの一部をHMAメモリに組み込むことが可能になった。
- デバイスドライバや常駐プログラムをUMBメモリに組み込むことが可能になった。

● DOS5との相違

- HIMEM.SYSが64Mバイトを超える拡張メモリに対応した。
- EMM386.EXEが拡張メモリ仕様のひとつであるDPMIに対応した。
- EMM386.EXEがEMSメモリとXMSメモリを一括して管理するようになった。
- ディスクキャッシュのSMARTDRVがCD-ROM・ネットワークドライブに対応したほか、ハードディスクやフロッピーディスクのライトキャッシュ（書き込みキャッシュ）が可能になった。
- CD-ROM用のMSCDEXコマンドが標準添付になった。
- FORMATコマンドなどが、1.44Mバイトのフロッピーディスクや128MバイトのMO（光ディスク）に対応した。
- 起動時にCONFIG.SYSやAUTOEXEC.BATの内容をスキップしたり、1行ずつ選択しながら実行することが可能になった。

● DOS3と DOS5/DOS6の違いは大きい

DOS3 にとっては、ハードディスクの管理領域が最大 128M バイトまでという制限は、現在ではかなり厳しいものになってきているのではないでしょうか。たとえば、500M バイトのハードディスクを購入した場合、DOS3 では最低でも4つの領域に区切らなければ利用できませんが、DOS5 や DOS6 であれば1ドライブとして利用できます。

また、DOS5/DOS6 では、常駐プログラムやデバイスドライバを UMB という領域に読み込むことができるため、その分、プログラムを実行するために必要なメモリを多く確保することができるのです。

これらのほかにも、コマンドの一部が拡張されたり、新規のコマンドが追加されたりしています。

●ユーティリティが充実した DOS6

基本的には、DOS5 と DOS6 では大きな仕様の違いはありません。HIMEM.SYS が、14.6M バイトを超えた拡張メモリも管理できるようになったので、それ以上のメモリを搭載したシステムなら乗り換えるほうがよいでしょう。

また、DOS6 では従来なら市販のソフトウェアが必要であった作業でも、たいていは付属のユーティリティで済ませてしまえるようになっています。なかでも、最もやっかいな作業と言われていた CONFIG.SYS や AUTOEXEC.BAT の最適化について、MemMaker というユーティリティを使って、誰にでも簡単に行えるようになりました。

DOS6 で付属になったユーティリティには、次のものがあります。

- **MemMaker**：CONFIG.SYS や AUTOEXEC.BAT の最適化を行う。
- **DoubleSpace**：ディスクを圧縮して、容量を拡大する。
- **Arcada Backup**：フルスクリーンのメニューで、ディスクのバックアップを行う。
- **Undelete／MWUNDEL**：ファイルを復活させる（DOS 版／Windows 版）。
- **Defrag**：不連続化したファイルの再配置を行う。
- **ウィルスバスター 98**：コンピュータ・ウィルスの検出を行う（DOS 版／Windows 版）。
- **SCANDISK**：ディスクの詳しい検査を行う。

・SEDIT　　　　：フルスクリーンでファイルの編集を行うスクリーンエディタ。
・HELP　　　　：フルスクリーンで、DOSの各コマンドに関する詳しい解説を行う。
・MAXLINK-LITE：2台のPC-98をケーブルでつないで、ファイルの転送を行う。

これらのうち、SEDITは『MIFES』（メガソフト）というエディタの機能縮小版です。また、MAXLINK-LITEは、『MAXLINK』（メガソフト）という商品名で専用ケーブルと共に販売されています。市販の『MAXLINK』のほうは、PC/AT互換機（DOS/V機）同士や、PC-98とPC/AT互換機の間で高速にファイル転送を行うことができます。ウィルスバスター98も、『ウィルスバスター』（リンク）という名の市販ソフト（98版・DOS/V版）があります。これらのユーティリティについては、とりあえず使ってみて、機能的に不足を感じたら、市販版を購入してもよいと思います。

DoubleSpaceと同じ働きを持つディスク圧縮ソフトでは、『DiskX II』（エー・アイ・ソフト）もよく知られています。『DiskX II』のほうは、ハードディスクだけではなく、フロッピーディスクやMO（光ディスク）も圧縮することができます。

その他、Arcada Backup、Undelete、Defrag、SCANDISKについても、同じ機能を持った市販のユーティリティが存在します。

DOS6購入時の注意

● DOS3ユーザーの注意点

すでに解説したように、DOS6では2Gバイトまでのハードディスクを1ドライブとして扱うことが可能になっています。DOS3からDOS6への移行時に、領域（パーティション）の変更を行うのもよいでしょう。ただし、領域を変更する前に、ハードディスク内にあるすべてのファイルのバックアップをとっておくことを忘れないでください。領域を解放すると、領域内にあったすべてのファイルが失われます。

なお、DOS3のBACKUPコマンドによってバックアップしたファイルは、DOS6に付属のQBACKUPコマンドでは復元できません。DOS6にもBACKUPコマンドがあるので、そちらを使ってください。

また、DOS3のFORMATコマンドで作成した領域に対しては、DOS6でも正常に認識されますが、DOS6のFORMATコマンドで作成した128Mバイトを超える領域に対しては、DOS3では認識されなくなってしまいます。DOS3のシステムが転送されたフロッピーディスクから起動する場合や、DOS6からDOS3に戻すような場合には、この点に十分注意してください。

● 2種類のセットがあるのに注意

DOS6のパッケージには、次のような種類があります。

- **最初から利用するためのフルセット**
- **特定のバージョンのためのアップグレード用セット**

アップグレード用セットの場合には、あらかじめ特定のバージョンのDOSがインストールされていないと、インストールできないことになっています。購入時には、十分確認するようにしてください。転送されるファイルの内容については、フルセットとアップグレード用セットとで違いはありません。

Step 2

インストール時の設定

●ポイント●

・インストール時に作成された CONFIG.SYS と AUTOEXEC.BAT は、最低限の設定になっている。
・インストール後に Windows 用ユーティリティを組み込むには、INSTWIN コマンドを実行する。

●解 説●

インストール時に作成されるCONFIG.SYSとAUTOEXEC.BAT

インストールを実行すると、CONFIG.SYS、AUTOEXEC.BAT が自動的に作成されます。その内容は、システムの構成や、インストール時の選択によって多少は異なりますが、おおよそ、次ページ図のようになっているはずです。

ここでは、インストール先には、A ドライブの DOS というディレクトリを指定しています。

◉ CONFIG.SYS の例

```
FILES=30
BUFFERS=10
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB /T=A:¥DOS¥EXTDSWAP.SYS
DEVICE=A:¥DOS¥SETVER.EXE
DEVICEHIGH=A:¥DOS¥PRINT.SYS /U
DEVICEHIGH=A:¥DOS¥RSDRV.SYS
DEVICEHIGH=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH,UMB
```

◉ AUTOEXEC.BAT の例

```
@ECHO OFF
PATH=A:¥DOS;A:¥
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
A:¥DOS¥SMARTDRV.EXE /X
MOUSE
DOSSHELL
MOUSE /R
```

この基礎編では、これらの設定を元に、内容について詳しく解説していきます。

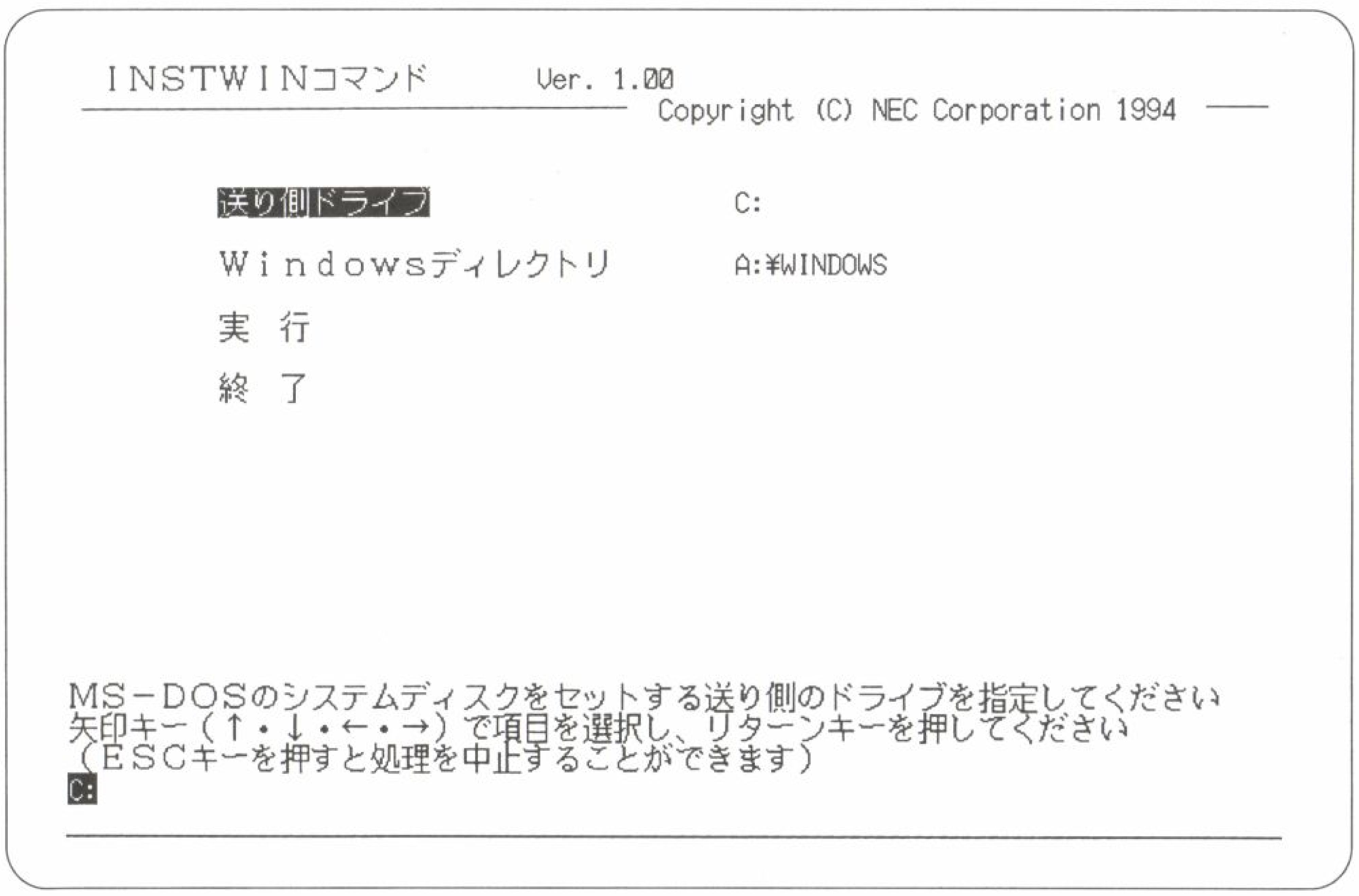
あとからWindows版ユーティリティを追加するには

MWUNDEL（Windows版のUNDELETEコマンド）、ウィルスバスター98、DoubleSpace情報の各Windows版ユーティリティは、DOS6をインストールする際、すでにWindowsがインストールされていると、自動的に組み込むためのプログラムが起動されます。

DOS6のあとからWindowsをインストールした場合には、次のようにINSTWINというコマンドを実行すれば組み込むことができます。

```
A:¥>INSTWIN
```

◉ INSTWINコマンドの起動画面



送り側ドライブに、システムディスクをセットするドライブ名を指定してください。Windowsがインストールされているディレクトリは、自動的に検索されるようになっています。

「実行」を指定したあとは、画面の指示に従ってください。

Step 3

CONFIG.SYS と AUTOEXEC.BAT のスキップとステップ実行

●ポイント●

・起動時に、CONFIG.SYS と AUTOEXEC.BAT の実行をスキップするには f•5 キーを押す。
・起動時に、CONFIG.SYS と AUTOEXEC.BAT を 1 行ずつ確認しながら実行するには f•8 キーを押す。
・起動時に、ディスク圧縮プログラムを組み込まずに、CONFIG.SYS と AUTOEXEC.BAT の実行をスキップするには CTRL ＋ f•5 キーを押す。
・起動時に、ディスク圧縮プログラムを組み込まずに、CONFIG.SYS と AUTOEXEC.BAT を 1 行ずつ確認しながら実行するには CTRL ＋ f•8 キーを押す。

●解 説●

CONFIG.SYS と AUTOEXEC.BAT の実行をスキップする

CONFIG.SYS と AUTOEXEC.BAT は、起動ドライブのルートディレクトリにあると、自動的に実行されるようになっています。記述方法を誤るなどして、実行の途中でシステムがストップしてしまう場合は、実行しないでシステムを起動させたり、それぞれの内容を 1 行ずつ実行させることもできるようになっています。

DOS6 で、CONFIG.SYS と AUTOEXEC.BAT を実行せずにシステムを起動するには、コピーライト表示が行われている約 2 秒の間に、

f•5 キー

を押してください。

●起動時のコピーライト表示

```
ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 6.20
Copyright (C) 1981,1994 Microsoft Corp. / NEC Corporation
```

ただし、CONFIG.SYS と AUTOEXEC.BAT をスキップしてしまうと、最低限の設定でシステムを起動することになります。一般のアプリケーションなどは、ほとんど起動できない環境だと思われますから、SEDIT コマンドなどを用いて、直ちに CONFIG.SYS や AUTOEXEC.BAT を修正しましょう。

ディスク圧縮プログラム(DoubleSpace)が組み込まれている場合、[CTRL]＋[f･5]キーを押すと、ディスク圧縮プログラムを組み込まずにスキップ起動することができます。

CONFIG.SYSとAUTOEXEC.BATのステップ実行

CONFIG.SYS と AUTOEXEC.BAT を完全にスキップしてしまうと、不便この上ないシステムになってしまいます。そこで、誤りのある行だけをスキップできるように、1行ずつ確認しながら実行するステップ実行の機能を利用するとよいでしょう。

ステップ実行を行うには、スキップする場合と同じく、コピーライト表示が行われている間に

[f･8]キー

を押してください。

ステップ実行では、個々の行について、Yes（実行）か No（スキップ）かを問い合わせてきます。[Y]キーまたは[N]キーを押してください。

[Y]/[N]キーの代わりに[ESC]キーを押すと、残りの行を連続して実行します。また、[f･5]キーを押すと、残りの行をすべてスキップします。

ディスク圧縮プログラム(DoubleSpace)が組み込まれている場合、[CTRL]＋[f･8]キーを押すと、ディスク圧縮プログラムを組み込まずにステップ実行することができます。

Step 4

CONFIG.SYS の基本コマンド①【BUFFERS】

●ポイント●

・BUFFERS はディスクバッファと先読みバッファの指定。
・SMARTDRV を使うなら、ディスクバッファは最低限に、先読みバッファは初期値の 0 にしておく。
・BUFFERS は HMA に収まる範囲で指定する。
・DoubleSpace を組み込む場合には、BUFFERS の値はさらに小さく設定する。

●解 説●

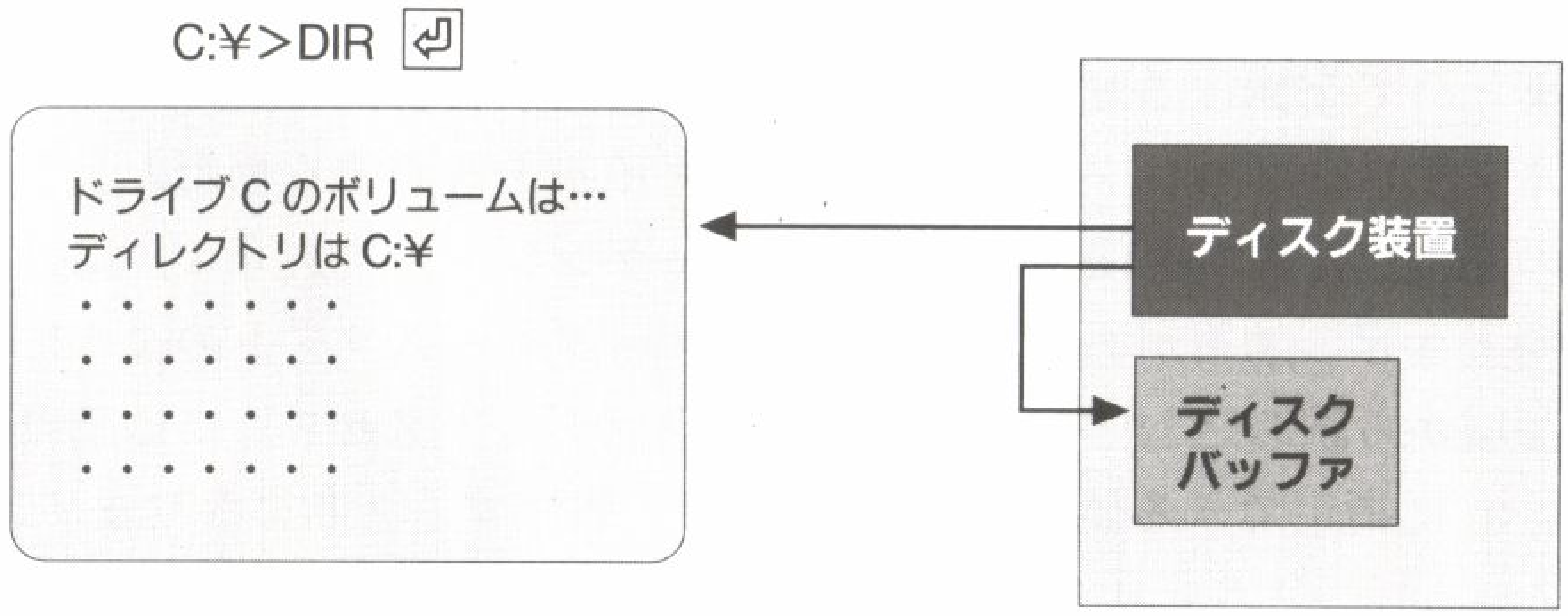
ディスクバッファとは

BUFFERS は、**ディスクバッファ**の容量を指定するコマンドです。**バッファ**（BUFFER）を訳すと「緩衝器」ですが、コンピュータ用語としては、情報を一時的に蓄えておくためのメモリのことになります。速度の異なる装置の間にバッファを入れることにより、速度差による処理時間の無駄を少なくする役目を果たすのです。

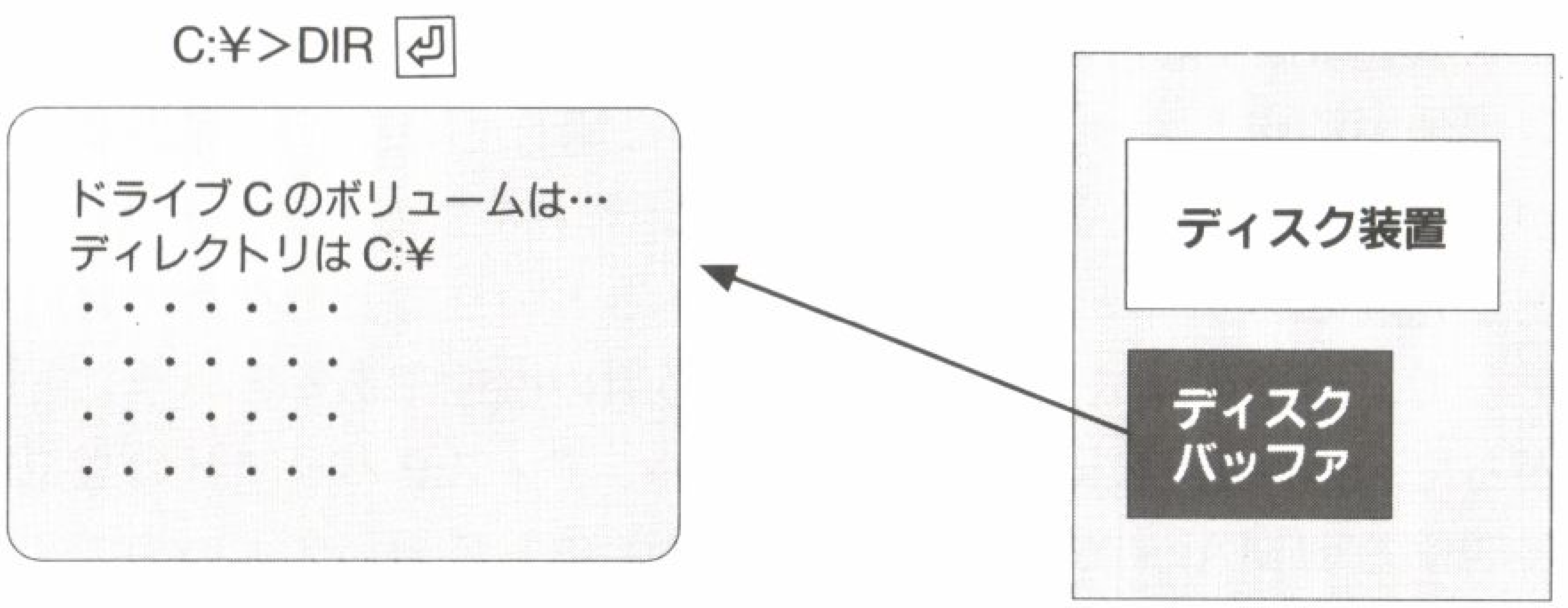
ディスクバッファは、CPU とディスク装置の間に設けられたバッファです。CPU からディスク装置にアクセス（読み書き命令）があった場合、ディスク装置から読み出すファイルの管理情報を含めたデータをバッファにも記憶しておき、2 度目に同じディスクを使う際は、バッファの情報を使って高速にアクセスします。

◉ディスクバッファのしくみ

①最初の読み出し時に、ディスクバッファへも記憶させておく



②同じ部分を読み出すなら、ディスクバッファのデータを用いる



BUFFERSの値と消費メモリの関係

BUFFERSコマンドは、CONFIG.SYS内で次の書式に従って記述します。

```
BUFFERS = n(,m)
```

●ディスクバッファの指定

nには、確保するディスクバッファの数を2〜63の範囲で指定します。640Kバイトのメモリを搭載したシステムや、ハイレゾリューションモードでの初期値は20で、それ以下のメモリになると徐々に減っていきます。しかし、現在では640Kバイト未満のメモリというシステムはありえないので、初期値が20と覚えておいていいでしょう。

●先読みバッファの指定

mは、**先読みバッファ**または**2次バッファ**と呼ばれる設定です。先読みバッファは、ディスク装置からの読み出し命令があった場合、連続した領域（セクタ）をまとめて記憶しておくためのバッファになります。

エディタやワープロソフトでテキストファイルを参照する場合のように、ディスクの連続した領域を使用する処理（シーケンシャルアクセス）には、処理速度を向上させる効果があります。一方、データベースソフトのように、あちこちの領域から読み込むことの多い処理（ランダムアクセス）には効果がありません。mに指定できる値は1〜8の範囲で、省略すると0になります。

SMARTDRVを使う場合の注意

DOS6やWindowsに付属しているSMARTDRVは、**ディスクキャッシュ**と呼ばれる機能を提供します。SMARTDRVに関しての詳しい解説はあとで行いますが、ここでは、SMARTDRVのようなディスクキャッシュは、ディスクバッファと先読みバッファの機能を含んでいるということを覚えておいてください。そして、今やディスクキャッシュは必須の機能であるということもです。

したがって、SMARTDRVを組み込むなら、ディスクバッファは最低限の値（次に説明するHMAメモリに収まる範囲内）にしておいてよいのです。その際、先読みバッファは指定してはいけません。

ディスクバッファはHMAの範囲内で

DOS6ではBUFFERSで設定したディスクバッファがHMAメモリに確保されるようになっています。ただし、HMAメモリに収まりきらない場合には、すべて基本メモリに確保されます。メモリ節約のため、ディスクバッファはHMAメモリの範囲内で設定するようにしましょう。消費メモリは、接続しているドライブの容量によって異なりますが、ディスクバッファ1つについて約1〜2Kバイトです。

HMAメモリに収まる範囲は、機種や、インストールされているプログラムによって異なってくるので、少しずつ設定値を増やしながらMEMコマンドによって確認してください。

HMA メモリからはみ出ると、「実行可能プログラム最大サイズ」が突然減るのでわかります。筆者が確認したところ、PC-9821Xa/C10W では 6 まで、PC-9821Lt/540A では 3 までなら HMA メモリに収まっていました。

BUFFERS を小さくすると、DOS が起動するまでの時間が遅くなるということはありますが、極端な違いはありません。それより、メモリを節約するほうがメリットが大きいと思います（もちろん、好みの問題ではありますが）。

◉ MEM コマンドによるメモリの確認

```
メモリの種類          合計   =   使用   +   空き
----------------    -------    -------    -------
ｺﾝﾍﾞﾝｼｮﾅﾙ             640K       106K       534K
ｱｯﾊﾟｰ ﾒﾓﾘ              82K        69K        13K
予約済み                0K         0K         0K
XMS ﾒﾓﾘ            23,470K    12,246K    11,224K
----------------    -------    -------    -------
全ﾒﾓﾘ              24,192K    12,421K    11,771K

全 1MB 以下ﾒﾓﾘ        722K       175K       547K

全 EMS ﾒﾓﾘ                     23,488K  (24,051,712 ﾊﾞｲﾄ)
空き EMS ﾒﾓﾘ                   11,680K  (11,960,320 ﾊﾞｲﾄ)

最大実行可能ﾌﾟﾛｸﾞﾗﾑｻｲｽﾞ          534K  (546,896 ﾊﾞｲﾄ)
最大空きｱｯﾊﾟｰﾒﾓﾘﾌﾞﾛｯｸ             6K  (6,544 ﾊﾞｲﾄ)
MS-DOS はハイメモリ領域に常駐しています.
```

ここをチェック（534K）

DoubleSpaceを組み込む場合の設定

DOS6に付属のディスク圧縮プログラム**DoubleSpace**を組み込む場合、HMAメモリに収まるBUFFERSの値はさらに小さくなります。それは、DoubleSpaceもHMAメモリを利用するようになっているからです。

DoubleSpaceを組み込む場合、HMAメモリの使われ方はちょっと複雑になっています。次の表は、PC-9821Xa/C10Wで1Gバイトのハードディスクを500Mバイト×2に分割している環境で、BUFFERSの値を少しずつ変更しながら、空きメモリの増減を調べた結果です。

BUFFERS	基本メモリの空き容量	HMAメモリの使用容量	HMAメモリ内
2	514592バイト	62バイト	①②③④
3	514592バイト	63バイト	①②③④
4	512464バイト	61バイト	①　③④
5	512464バイト	62バイト	①　③④
6	509344バイト	60バイト	①②　④

①：DOSシステムの一部　②：COMMAND.COMの一部
③：ディスクバッファ　④：DoubleSpaceの一部

この結果を見ると、BUFFERSの値を3以下にしたときが最も多くの空きメモリを確保できています。それは、4〜5にするとCOMMAND.COMがHMAメモリを使わなくなり、6以上にするとディスクバッファが基本メモリに確保されるようになるからです。

最適な値はシステム環境によって異なりますから、実際にBUFFERSの値を1〜8程度の範囲で変えながら起動してみて、MEMコマンドの実行結果を比較してみてください。たとえば、PC-9821Lt/540AでDoubleSpaceを組み込んだ場合には、1に設定しないとHMAメモリからはみ出てしまいました。

Step 5

CONFIG.SYS の基本コマンド②【FILES, FCBS】

●ポイント●

・FILES はオープンできる最大ファイル数の指定。
・Windows を使うなら FILES は 30 以上にする。
・LAN やデータベースソフトは指定に注意。
・FCBS は最低値の 1 でよい。

●解 説●

FILES は「30」がお勧め

MS-DOS では、ファイルを利用するための宣言として、**オープン**という処理を行います。そして、必要がなくなれば**クローズ**という処理を行うようになっています。

FILES コマンドは、同時にオープン可能なファイルの最大数の指定です。CONFIG.SYS 内で次の書式に従って記述します。

```
FILES = n
```

n は、8～255 の範囲で指定します。FILES を指定しなければ、初期値の 8 で使用することになりますが、それではとても満足な環境とは言えません。最低でも 15 程度は必要と考えてください。Windows では、FILES を最低でも 30 にするようにとの指示があります。

ディスクバッファほどではありませんが、FILESの値によっても基本メモリは消費されていきますから、必要以上に多く設定するのはメモリの無駄使いというものです。DOS環境、Windows環境共に、30を指定しておけばよいでしょう。

なお、LAN環境で利用したり、データベースソフトや言語ソフトを使う場合など、特に多くのファイルをオープンする必要がある場合には、それぞれのマニュアルで指定された値にしてください。インストールプログラムによって、自動的に書き換えられることもあります。

FCBSは最低値でよい

ファイルに関するコマンドとしては、ほかにも**FCBS**があります。

```
FCBS = n          (n:1～255)
```

初期のプログラムには、**FCB**(**File Control Block**:ファイル制御ブロック)を使ってファイルを操作するものがありました。しかし、現在のプログラムでFCBを使っているものは皆無と言ってよいため、FCBを確保する必要はまったくありません。

FCBSコマンドを記述する必要はありませんが、初期値が4になっているため、最低値の1に変更しておけば、ほんの少しだけメモリの節約になります。

```
FCBS = 1
```

Step 6

CONFIG.SYS の基本コマンド③【LASTDRIVE】

●ポイント●

・LASTDRIVE は使用する最大ドライブ数の指定。
・仮想ドライブやネットワークドライブの使用時には設定に注意。

●解 説●

基本的には LASTDRIVE の設定は不要

LASTDRIVE コマンドでは、使用するドライブの最大数を指定します。

```
LASTDRIVE = x
```

x に指定するのは A~Z までのアルファベットで、たとえば、7 台のドライブを使用するなら G になります。つまり、MS-DOS では最大で 26 のドライブが使用可能というわけです。

LASTDRIVE の初期値は E、つまり、5 台分のドライブ名を確保します。もし、それより多くのドライブが接続されていると、実際に使用されているドライブ数に合わせて自動的に設定されますから、通常、このコマンドを記述する必要は特にありません。

あとからドライブ数が増えるなら設定が必要

LASTDRIVE で設定するドライブには、ハードディスク、RAM ディスク、CD-ROM ドライブ、MO ディスクなどはもちろんのこと、SUBST コマンドで割り当てた**仮想ドライブ**、ディスク圧縮プログラムで設定された**圧縮ドライブ**、LAN 環境で利用する**ネットワークドライブ**などがすべて含まれます。

これらのうち、仮想ドライブやネットワークドライブなど、システムの起動後にドライブ数が増える場合には、あらかじめ LASTDRIVE コマンドで最大数を増やしておくようにしてください。

また、CD-ROM ドライブを利用する場合には、CD-ROM ドライブに割り当てられる分も含めて LASTDRIVE コマンドで設定しておかないと、ドライブが認識されないことがあります。

LASTDRIVE コマンドの設定値に従って、若干ですが基本メモリも消費されていくため、必要がなければ余分なドライブ数は確保しないでおきましょう。

ただし、ネットワークや CD-ROM のドライバソフトのインストール時、インストールプログラムによって、

```
LASTDRIVE=Z
```

のように最大値に設定された場合には、その設定で利用するほうがよいでしょう。

Step 7

CONFIG.SYS の基本コマンド④【SHELL】

●ポイント●

- SHELL はコマンドインタプリタの場所を指定。
- コマンドインタプリタが起動ドライブのルートディレクトリになければ再読み込み先の指定が必要。
- COMMAND.COM には/P スイッチが必要。
- 環境変数を多く使うなら/E スイッチで環境エリアを広げておく。
- 再読み込み先は環境変数 COMSPEC にセットされる。

●解 説●

コマンドインタプリタとは

SHELL コマンドでは、MS-DOS 環境下においてユーザーからのコマンドを解釈して、それを実行させるためのプログラムである**コマンドインタプリタ**の指定を行います。

通常は、COMMAND.COM がコマンドインタプリタとして用いられます。コマンドインタプリタが起動ドライブのルートディレクトリにある場合には、必ずしも SHELL コマンドによって指定しておく必要はありません。

たとえば、COMMAND.COM が A ドライブの DOS というディレクトリにあるなら、

```
SHELL = A:¥DOS¥COMMAND.COM A:¥DOS /P
```

と記述してください。

COMMAND.COMのあとに「A:¥DOS」とあるのは、COMMAND.COMの再読み込み先の指定です。

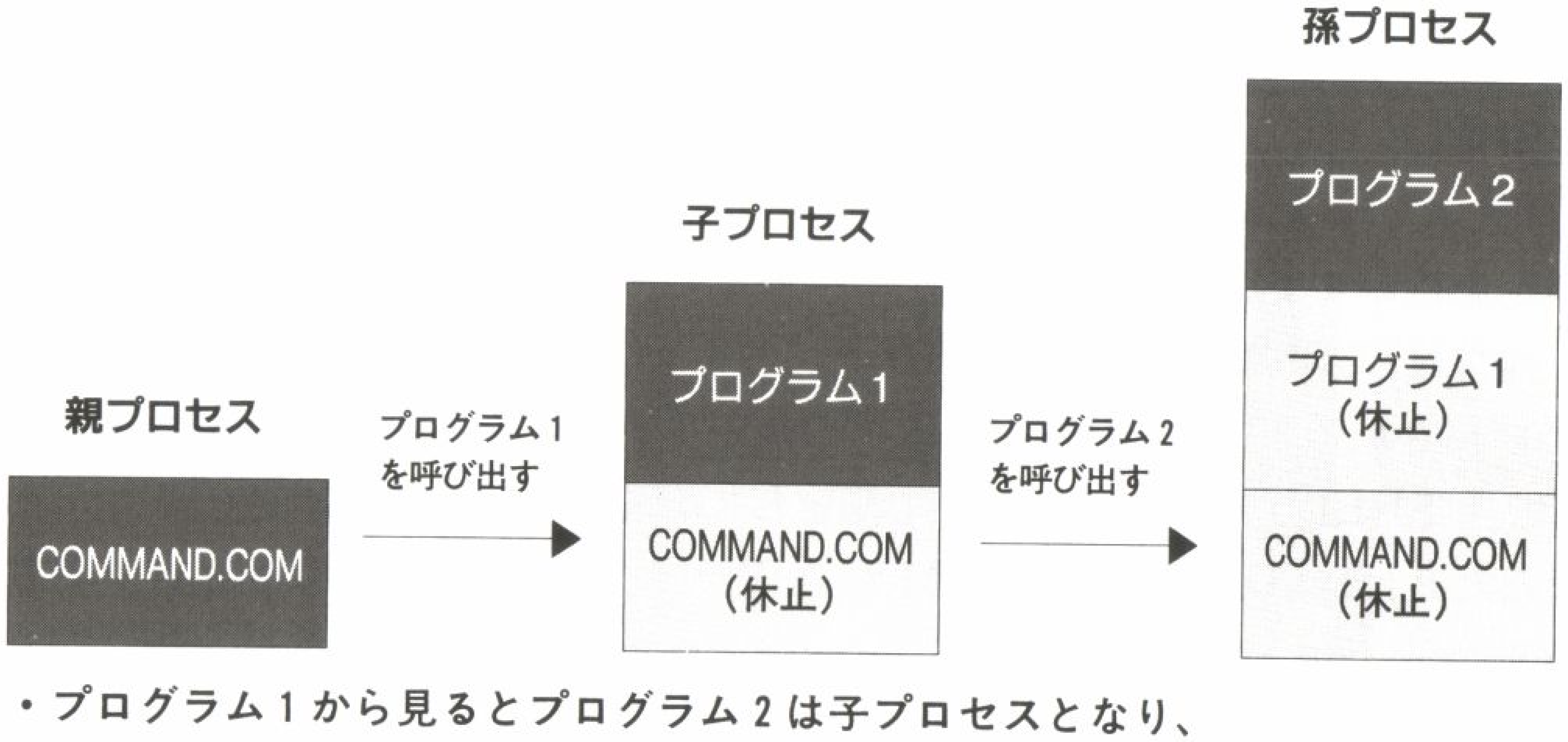
MS-DOSが起動すると、COMMAND.COMは基本メモリに常駐しますが、大きなプログラムを実行した場合は、その常駐部分のメモリが書き換えられてしまうことがあります。その際、SHELLコマンドで指定した再読み込み先からCOMMAND.COMを探して再実行するのです。コマンドインタプリタが起動ドライブのルートディレクトリにない場合には、必ず再読み込み先を指定しておいてください。

COMMAND.COMの/Pスイッチ

SHELLコマンドでCOMMAND.COMを指定する際には、必ず/Pスイッチを付けるものと考えてください。これから、その理由を説明します。

MS-DOSでは、実行中のプログラムから別のプログラムを呼び出して実行することができます。最初に実行していたプログラムは**親プロセス**となり、次に実行されたプログラムは**子プロセス**、その次は**孫プロセス**……となっていくのです。そして、別のプログラム内から実行したプログラムを終了するには、EXITコマンドを用います。

◉プログラム呼び出しの様子



・プログラム1から見るとプログラム2は子プロセスとなり、
プログラム2から見るとプログラム1は親プロセスになる

さて、MS-DOSのコマンドラインから実行したプログラムは、実はすべてコマンドインタプリタ（=COMMAND.COM）が呼び出すしくみになっています。そのため、コマンドインタプリタが**EXIT**コマンドで終了してしまうことがあると、何もプログラムが実行できず、システムが機能しなくなってしまうのです。COMMAND.COMに付ける/Pスイッチは、EXITコマンドによる終了を禁止するためのスイッチです。

もうひとつの理由として、/Pスイッチを付けておかないと、CONFIG.SYSに続けてAUTOEXEC.BATが実行されなくなってしまうのです。

環境エリアを広げる/Eスイッチ

AUTOEXEC.BAT内で**SET**コマンドを用いて設定する**環境変数**は、すべて**環境エリア（環境領域）**に保存されます。初期設定では、環境エリアのサイズは256バイトになっていますが、もしも環境変数が多くなって、

```
環境変数のための領域が足りません.
```

というエラーメッセージが表示されたら、COMMAND.COMの/Eスイッチを使って、領域を広げるようにしてください。

◉ COMMAND.COMの/Eスイッチによる環境領域の指定

```
SHELL = A:¥DOS¥COMMAND.COM A:¥DOS /P /E:バイト数
```

/Eスイッチで指定できるのは160〜32768バイトの範囲で、たとえば、512バイトにするなら、「/E:512」とします。値は16の倍数を指定するようにしてください（端数は切り上げ）。たいてい512もあれば足りるはずです。

●環境領域のサイズを調べる

環境変数の大きさを調べるには、SET コマンドの結果をファイルにリダイレクトします。F ドライブに SET_DATA というファイルで出力するなら、

```
A:¥>SET >F:¥SET_DATA
```

と実行してください。

◉ SET コマンドの実行画面

```
A:¥>SET
PATH=A:¥DOS;A:¥;A:¥DOS;A:¥JSLIB;A:¥WINDOWS;A:¥BAT;A:¥TOOL;A:¥MYT
COMSPEC=A:¥COMMAND.COM
HSBDIR=CONAUT
HSBEXT=TMP
HSBM=A:¥TOOL
SOUND=A:¥SB16
BLASTER=Ad2 I10 D3
MIDI=SYNTH:1 MAP:E
HD1=A:
HD2=B:
RD1=E:
RD~=E:
BOOTDRV=A:
SYSDRV=A:
LASTDRV=E:
DOSVER=6.20
TEMP=E:¥
PROMPT=$P$G
```

出力されたファイルのサイズを DIR コマンドなどでチェックすれば、環境変数のサイズがわかります。

◉ DIR コマンドによるサイズの確認

```
A:¥>DIR

 ドライブ E: のボリュームラベルは MS-RAMDRIVE
 ディレクトリは E:¥
                                 ──────────────ここをチェック
SET_DATA             289 95-06-21    7:51
ZU7                    0 95-06-21    7:52
         2 個              289 バイトのファイルがあります
                     6,262,784 バイトの空きがあります
```

なお、先に説明した/P スイッチや/E スイッチは、すべて COMMAND.COM に対するスイッチであって、SHELL コマンドにはスイッチ類はありません。

環境変数 COMSPEC との関係

SHELL コマンドで指定されたコマンドインタプリタの再読み込み先は、自動的に環境変数 COMSPEC にセットされます。

たとえば、

```
SHELL = A:¥DOS¥COMMAND.COM A:¥DOS /P
```

と設定してあれば、

```
COMSPEC=A:¥DOS¥COMMAND.COM
```

と設定されます。

したがって、SHELL コマンドの代わりに AUTOEXEC.BAT 内で、

```
SET COMSPEC=A:¥DOS¥COMMAND.COM
```

のように指定することも可能です。ただし、あとから環境エリアを広げることもできるように、SHELL コマンドを使っておくほうがよいと思います。

Step 8

CONFIG.SYSの基本コマンド⑤【DOS】

●ポイント●

・DOSシステムの一部をHMAメモリに組み込むならDOS＝HIGH。
・UMBメモリをDOSに管理させるならDOS＝UMB。

●解 説●

DOSコマンドは必須の設定

DOSコマンドは、HMAメモリにDOSシステムの一部を組み込む指定を行い、DOSにUMBメモリの管理を命じるコマンドです。十分な基本メモリを確保するためには、これらの指定は必要不可欠と言えます。

DOSコマンドは、次のように記述してください。

```
DOS = HIGH   HMAメモリにDOSシステムの一部を組み込む
DOS = UMB    DOSがUMBメモリを管理する
```

また、両方を同時に指定するなら、次のようにまとめて記述できます。

```
DOS = HIGH,UMB
```

HIGHとUMBの順番を入れ替えて「DOS＝UMB,HIGH」としても同じ意味になります。これらのほか、DOSコマンドにはHMAメモリにシステムの一部を組み込まないLOWと、UMBメモリを管理しないNOUMBというパラメータもありますが、いずれも初期設定なので、覚えておく必要はないでしょう。

なお、HMAメモリとUMBメモリを使用するためには、HIMEM.SYSなどの拡張メモリマネージャを組み込む必要があります。

Step 9

CONFIG.SYS の基本コマンド⑥【SET】

●ポイント●

・SET では環境変数を設定する。
・従来の DOS との互換性を考えるなら、環境変数は AUTOEXEC.BAT 内で設定するほうがよい。

●解 説●

DOS6 で加わった SET コマンド

DOS6 では、**SET** コマンドを用いて CONFIG.SYS 内で**環境変数**が設定できるようになっています。

```
SET 環境変数=文字列
```

たとえば、「TEMP」という環境変数に、「F:¥」という文字列を割り当てるなら、

```
SET TEMP=F:¥
```

と記述してください。このとき、「=」の前後にスペースを入れてはいけません。スペースも含めた文字列と判断されてしまうからです。

CONFIG.SYS の SET コマンドでは、環境変数、「=」、文字列の合計で 128 バイトまでが認識されます。同じ SET コマンドを AUTOEXEC.BAT で用いると、環境変数、「=」、文字列の合計が 123 バイトまでしか認識されませんから、5 バイト分だけ長い文字列が設定可能ということになります。

環境変数はAUTOEXEC.BAT内で設定するのが普通

DOS6より前のMS-DOSでは、CONFIG.SYS内でSETコマンドを用いることはできませんでした。また、同じDOS6でも、DOS/V版にはCONFIG.SYS内のSETコマンドはサポートされていません。

これらのことを考えると、CONFIG.SYS内でSETコマンドを用いることは、ちょっと特殊な使い方になってしまうように思えます。CONFIG.SYSで設定するメリットはあまりなさそうなので、どうせなら、AUTOEXEC.BAT内で記述するほうがよいのではないでしょうか。本書では、環境変数はAUTOEXEC.BAT内で記述するものとして解説していきます。

SETコマンドの使い方や、環境変数の意味については、「Step25 AUTOEXEC.BATの基本コマンド② 【SET】」で詳しく解説します。

Step 10

CONFIG.SYS の基本コマンド⑦【REM,;】

●ポイント●

・行の先頭に REM あるいはセミコロン(;)を記述すると、その行はコメント行として無視される。

●解 説●

コメント行は無視される

CONFIG.SYS のなかで、先頭に REM という文字列か、あるいはセミコロン（;）があると、その行は無視されるようになっています。わかりやすくするために途中でコメント（解説）を入れたり、同じコマンドで複数の設定を使い分けたりする際に利用するとよいでしょう。

◉コメントの例

```
;WINDOWSを使うための設定
REM FILES=40
```

セミコロン（;）のあとには、スペースを入れても入れなくてもよいのですが、REM コマンドのあとには、必ず半角スペースを入れるようにしてください。

Step 11

CONFIG.SYS の基本コマンド⑧【?】

●ポイント●

・CONFIG.SYS コマンドの直後に？を付けておくと、起動時に、その行を実行するかどうかの問い合わせがある。

●解 説●

起動環境の選択や設定のチェックに便利

CONFIG.SYS のコマンドの後ろに「?」という記号を入れておくと、その行を実行してもよいかという問い合わせが行われるようになります。コピーライトが表示されている間に f・8 キーを押した場合には、すべての行について問い合わせが行われますが、「?」を用いると、一部の行だけを対象にすることができるのです。

たとえば、次のように記述しておけば、ディスクバッファの値を 30 または 50 に設定することができます。

```
BUFFERS? = 30
BUFFERS? = 50
```

CONFIG.SYS コマンドの直後に記述する点を間違えないようにしてください。この例では、BUFFERS コマンドと「?」の間にスペースを入れるとエラーになってしまいます。ただし、「?」と「=」の間にスペースを入れるのはかまいません。

後述する常駐プログラムやデバイスドライバの組み込みを選択したい場合に用いても便利です。

◉ PRINT.SYS の組み込みを確認する例

```
DEVICEHIGH? = A:¥DOS¥PRINT.SYS
```

EMM386.EXE のように、誤った設定をすると起動できなくなるようなプログラムの設定を行う際には、確実に起動できる内容の行と、試しに起動してみたい内容の行を「?」を付けて記述しておけば安全です。

Step 12

デバイスドライバとは

●ポイント●

・標準デバイスドライバには、CON、AUX、COM1、COM2、PRN、LPT1、LPT2、NUL、CLOCK、A:～Z:がある。
・デバイスドライバには、キャラクタ型とブロック型とがある。
・キャラクタ型ドライバと同じ名前のファイルやディレクトリは作成できない。

●解 説●

標準デバイスドライバ

PC-9800 シリーズは、本体内部の専用スロットや、背面にある汎用拡張スロットを用いて、機能を拡張できるようになっています。そのため、一度購入したシステムでも、ある程度は技術の進歩に取り残されることなく使い続けることができるのです。たとえば、CD-ROM ドライブ、ビデオ画像の取り込み、高度なサウンドなど、豊富なオプション機器が用意されています。

これは、ハードウェア面だけではなく、ソフトウェア面でも同じことが言えます。MS-DOS には、ディスプレイ、プリンタ、ディスク装置などを操作するための基本的な機能が**デバイスドライバ**（**ドライバ**、**デバドラ**などと略されます）として用意されていて、そのデバイスドライバを置き換えるだけで、最新の機器に対応することができるのです。

標準で用意されているデバイスドライバと、それぞれの初期設定は次のとおりです。

◉ DOS6の標準デバイスドライバ

デバイス名	意　味	初期設定
CON	標準入出力	キーボードとディスプレイ
AUX	補助入出力	RS-232C インターフェイスを用いた入出力
PRN	プリンタ出力	プリンタ
NUL	ダミー出力	ダミー出力
CLOCK	システムクロック	システムクロック
A:～Z:	補助記憶装置入出力	ディスク装置

NUL というデバイスドライバは特殊なもので、どこにも出力させたくない場合に用います。たとえば、AUTOEXEC.BAT 内で TEST.COM という常駐プログラムを実行する場合に、起動時のメッセージを表示しないようにするには、

```
TEST >NUL
```

のように記述しておきます。このテクニックを用いると、画面への出力をしない分だけ、起動時間が短くなります。

デバイスドライバの種類

デバイスドライバは、**キャラクタ型**と**ブロック型**の2種類に分けることができます。標準デバイスドライバのうち、A:～Z:までがブロック型で、ほかはすべてキャラクタ型になります。

キャラクタ型のデバイスドライバには、次のような特徴があります。

- 半角8文字までの名前を指定できる。
- データを1文字（キャラクタ）ずつ扱う。
- 同じ名前のデバイスドライバを組み込むと、先に組み込んであったものは無効となり、あとから組み込んだものが有効となる。

これに対して、ブロック型は次の特徴をもっています。

- A:〜Z:までの名前を用いる。
- 組み込んだ順番に A:から割り当てる。
- データをブロック単位で扱う。
- ランダムアクセスが可能。

以上の特徴からわかるように、ブロック型はディスク関連のために用いられ、そのほかのプリンタ、マウス、RS-232C による入出力などは、すべてキャラクタ型になります。

標準デバイスドライバのキャラクタ型はファイルとして管理

MS-DOS では、**リダイレクト**を使ってコマンドの実行結果をファイル化することができます。

◉ MEM コマンドの結果を MEM.DAT に出力する

```
A:¥>MEM >MEM.DAT
```

また、同じリダイレクト記号(**>**)を用いて、プリンタに出力することも可能です。

◉ MEM コマンドの結果をプリンタに出力する

```
A:¥>MEM >PRN
```

これは、キャラクタ型の標準デバイスドライバがファイルと同様に管理されているからです。そのため、標準デバイスドライバと同じ名前のファイルやディレクトリは作成することができません。

Step 13

デバイスドライバの組み込み方【DEVICE,DEVICEHIGH】

●ポイント●

・デバイスドライバはDEVICEコマンドで基本メモリに、DEVICEHIGHコマンドでUMBメモリに組み込む。
・UMBメモリに組み込むには、常駐サイズよりも大きな空きが必要。
・ギリギリのUMBメモリに組み込もうとすると、ハングアップすることもある。

●解 説●

CONFIG.SYSにおけるデバイスドライバの組み込み

デバイスドライバは、主にCONFIG.SYS内で組み込むようになります。キャラクタ型の場合には、起動後に組み込む場合もありますが、ブロック型の場合には、基本的にCONFIG.SYS内に限られると考えてください。

CONFIG.SYS内では、**DEVICE**コマンドを用いると基本メモリに組み込まれ、**DEVICEHIGH**コマンドを用いるとUMBメモリに組み込まれます。

◉ DEVICE(DEVICEHIGH)コマンドの書式

```
DEVICE = デバイスドライバ (オプション)
DEVICEHIGH = デバイスドライバ (オプション)
```

DEVICEHIGH コマンドを用いるには、その前に拡張メモリマネージャによってUMBメモリが有効になっていなければなりません。DOSに付属のマネージャを用いるなら、次の条件が必要になります。

- 先に HIMEM.SYS が組み込んであること。
- 先に EMM386.EXE で UMB メモリを割り当ててあること。
- DOS = UMB を記述すること。

DEVICEHIGH コマンドでUMBメモリに組み込もうとした場合、デバイスドライバの組み込みに十分な空きメモリがなければ基本メモリに組み込まれるようになります。一般に、デバイスドライバの組み込みには、常駐後のサイズよりも大きなメモリが必要になるため、常駐サイズからだけでは、UMBメモリに収まるかどうかは判断できません。

●組み込み時にハングアップした場合

組み込み時に必要なメモリよりも、UMBメモリの空きがほんの少しだけ小さいような場合には、組み込み時にシステムがストップ（ハングアップ）してしまうこともあるので注意が必要です。

万一、組み込み時にストップしてしまったら、起動時に[f•8]キーを押してCONFIG.SYSを1行ずつ実行して、どこが失敗の原因になっているか確認してください。それから、リセット後に再び[f•8]キーでステップ実行して、問題の行を[N]キーでスキップしてしまいます。起動後、CONFIG.SYSの該当部分をDEVICEコマンドに改めてください。

なお、DOS6のDEVICEHIGHコマンドには、UMBメモリの位置（ブロック）を指定するオプションも用意されていますが、それについては「Part3 応用編」で解説します。

デバイスドライバの拡張子

多くのデバイスドライバは、拡張子に「～.SYS」や「～.DRV」が用いられています。しかし、拡張子が「～.EXE」または「～.COM」になっているデバイスドライバもありますから、拡張子だけで判断することはできません。

Step 14

常駐プログラムとは

●ポイント●

・常駐プログラムは、実行後メモリ内に残って、決まったタイミングで起動する。
・デバイスドライバと常駐プログラムは、共にメモリ内に常駐するが、組み込み方法やシステム上の扱いが異なる。
・基本メモリに常駐させる場合には空きメモリに注意。

●解 説●

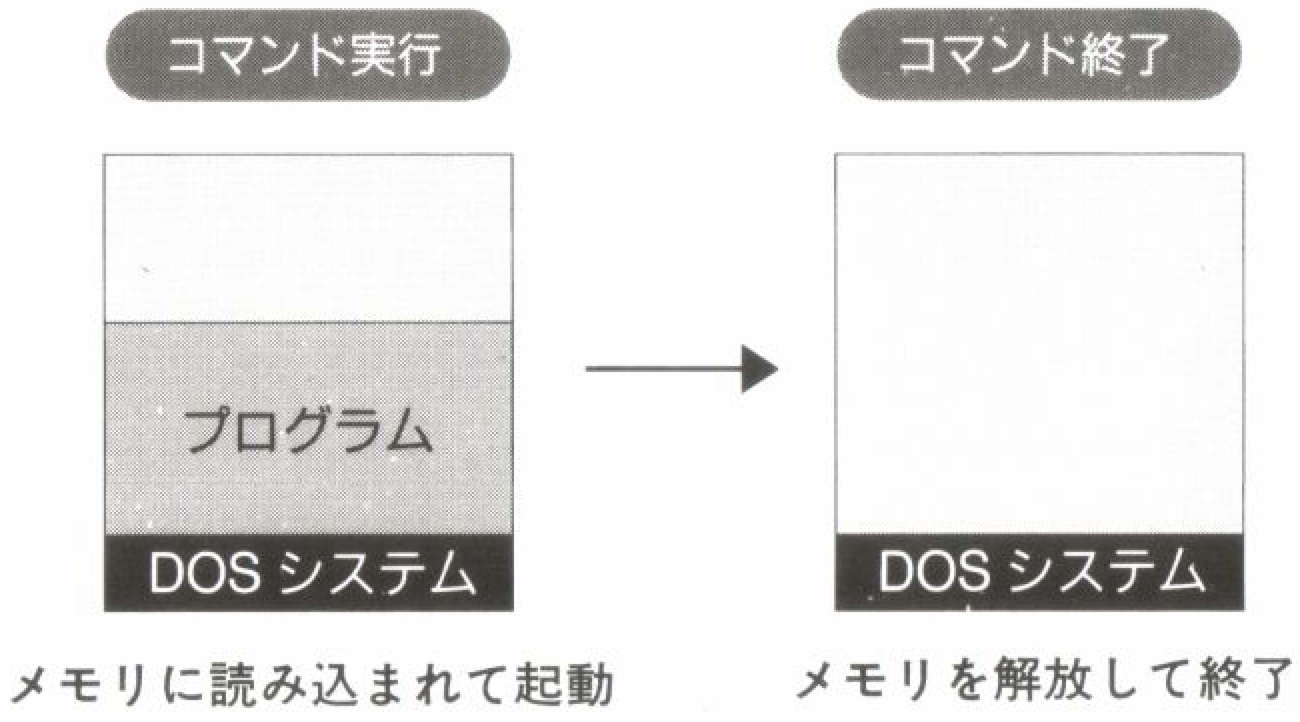
常駐プログラムのしくみ

通常のプログラムは、コマンドとして実行すると直ちに起動して、終了するとメモリをすべて解放します。しかし、**常駐プログラム**または**TSR**(terminate and stay resident program：常駐終了型プログラム）と呼ばれるタイプのプログラムは、コマンドとして実行しても起動はせずにいったん終了します。しかし、プログラム自体はメモリ内に残って、特定のタイミングでいつでも動作するように待機した状態になります。

常駐プログラムが起動するタイミングはさまざまですが、起動する条件のことを**トリガー**（trigger）と呼びます。

◉一般のプログラムと常駐プログラム

・一般のプログラム



・常駐プログラム



常駐プログラムとデバイスドライバとの違い

メモリに常駐するという意味では、デバイスドライバも同様です。しかし、デバイスドライバと常駐プログラムとでは、次のような違いがあります。

- デバイスドライバはプログラムとして実行するのではなく、DEVICE(HIGH)コマンドでメモリに常駐させるようになっている。
- 常駐プログラムがDOSの管理下にはないのに対して、デバイスドライバは常にDOSが管理している。
- 常駐プログラムは常駐を解除することが容易にできるが、デバイスドライバは基本的にできないようになっている。

デバイスドライバでも、MS-DOSに付属のADDDRV/DELDRVコマンドを用いて起動後に常駐させたり、常駐を解除することもできますが、それは特殊な使い方と言えます。

ただ、デバイスドライバの形式で組み込まれても、実は、単なる常駐プログラムとして機能するプログラムであったり、その逆に、常駐プログラムのスタイルになっていても、デバイスドライバとして働くようなプログラムであったりもします。見かけで判断することは難しいのですが、とにかく、どの方法でメモリに常駐させればよいのかということだけは覚えておきましょう。

常駐プログラムの実行は空きメモリと相性に注意

常駐プログラムは、UMBメモリや基本メモリ内に居座りますから、通常のプログラムのような感覚で実行してはいけません。常駐プログラムを実行するたびに、空きメモリが少なくなっていくのです。うっかりして同じプログラムを実行してしまっても、エラーにならずにダブルで常駐してしまうこともあります。

特に、基本メモリに常駐させる場合には、空きメモリとよく相談してからにしてください。アプリケーションによっては、空きメモリが少ないと起動できないものもあるからです。常駐プログラムは、用が済んだらなるべく常駐を解除しておくように心掛けてください。

それから、常駐プログラムを常駐させた状態で一般のアプリケーションを起動した場合や、複数の常駐プログラムを常駐させた場合には、プログラム同士の相性が悪く、何かの不具合が起きることがあります。そうした場合、相性の悪さを解消することは難しいため、常駐プログラムの常駐を解除してから、再び実行するようにしてください。

Step 15

常駐プログラムの組み込み方【LOADHIGH(LH),INSTALL, INSTALLHIGH】

●ポイント●

・常駐プログラムを組み込むには、コマンドラインから実行するか、AUTOEXEC.BAT内で記述する。
・LOADHIGH(LH)コマンドを使うと、UMBメモリに常駐させることができる。
・常駐プログラムによっては、CONFIG.SYS内でINSTALL(HIGH)コマンドを用いて組み込むことができる。
・起動後に常駐を解除したいプログラムはINSTALLコマンドを使わない。

●解 説●

常駐プログラムを常駐させる方法

常駐プログラムは、基本的にコマンドラインから実行すると、そのままメモリに常駐するようになっています。たとえば、MOUSE.COMという常駐プログラムであれば、

```
A:¥>MOUSE
```

のように実行すればよいのです。

AUTOEXEC.BAT内で記述しておく方法もあります。

```
MOUSE
```

どちらの場合でも、**LH** コマンド(**LOADHIGH** コマンドの略)を用いて UMB メモリに常駐させることができるプログラムもあります。

A:¥>LH MOUSE (コマンドラインから実行)
LH MOUSE (AUTOEXEC.BAT内に記述)

プログラムによっては、UMB メモリに常駐させることができなかったり、常駐させると正常に動作しなくなる場合があるため、マニュアルで確認するか、あらかじめ試しておく必要があります。

常駐プログラムの常駐を解除して、メモリを解放するには、特定のオプションを指定して実行すればよいはずです。オプションについては、それぞれのマニュアルやヘルプメッセージで確認してください。ただし、常駐を解除できないものもあります。

CONFIG.SYS 内での組み込み

常駐プログラムによっては、CONFIG.SYS 内で **INSTALL** コマンドまたは **INSTALLHIGH** コマンドを用いて組み込むことが可能なものもあります。INSTALLHIGH コマンドを用いると、UMB メモリに常駐させることができます。

◉ INSTALL(INSTALLHIGH)コマンドの書式

INSTALL = 常駐プログラム
INSTALLHIGH = 常駐プログラム

INSTALL(HIGH)コマンドを用いると、AUTOEXEC.BAT 内で組み込むよりも若干のメモリの節約になります。ただし、この方法はすべての常駐プログラムで利用できるわけではありません。安全のためには、コマンドラインから実行するか、AUTOEXEC.BAT 内で記述しておくほうがよいと思います。

また、INSTALL コマンドは起動後に常駐を解除しないプログラムに対して用いてください。たとえば、マウスドライバの MOUSE.COM を INSTALL コマンドで常駐させると、常駐が解除できなくなります。

なお、MS-DOS 6.2 のマニュアルや HELP コマンドには INSTALLHIGH コマンドの説明がありませんから、INSTALLHIGH は隠しコマンドということになると思います。メーカーのサポートの範囲外ということです。

Step 16

拡張メモリマネージャの組み込み【HIMEM.SYS,EMM386.EXE】

●ポイント●

- DEVICEHIGH コマンドを用いる前に、HIMEM.SYS と EMM386.EXE を組み込んでおく。
- HIMEM.SYS は EMM386.EXE より前に組み込む。
- 起動時のメモリチェックを省くには、HIMEM.SYS に/TESTMEM:OFF を指定する。
- EMM386.EXE の/T スイッチは、DOS シェルでタスクスワップ機能を利用していて、何か不具合が発生した場合にのみ指定するとよい。
- EMM386.EXE はプログラムの要求に応じて XMS メモリまたは EMS メモリを割り当てるため、EMS メモリのサイズを指定する必要はない。

●解 説●

拡張メモリマネージャ HIMEM.SYS

MS-DOS では、1M バイトを超えるメモリは**拡張メモリ**（**EMB メモリ**）として特別なプログラムによって管理しなくてはなりません。そのプログラムのことを**拡張メモリマネージャ**といいます。DOS6 には、拡張メモリマネージャとして **HIMEM.SYS** というデバイスドライバが付属になっています。

HIMEM.SYS によって管理される拡張メモリというのは、具体的には XMS 規格によるメモリ（**XMS メモリ**と **HMA メモリ**）を指しています。そのため、HIMEM.SYS を **XMS メモリマネージャ**とも呼びます。

HIMEM.SYSは、XMSメモリを利用するデバイスドライバより前で、DEVICEコマンドによって組み込む必要があります。

```
DEVICE = A:¥DOS¥HIMEM.SYS
```

HIMEM.SYSには、いくつかのオプションが用意されています。しかし、通常はオプションを付ける必要はありません。

HIMEM.SYSで有効にしたHMAメモリに、DOSのシステムの一部を入れるために、

```
DOS = HIGH
```

の記述を忘れないでください。

なお、MS-DOSに付属のHIMEM.SYSには、起動時に拡張メモリの信頼性をテストする機能があります。信頼性の低いメモリが見つかると、エラーメッセージが表示されますから、その場合には、本体（あるいはメモリ）を購入した店で、チェックを受けてください。

次のように記述しておくと、テストは行われなくなり、起動時間が短縮されます。

```
DEVICE = A:¥DOS¥HIMEM.SYS /TESTMEM:OFF
```

毎回チェックを行うほうが安全なのですが、大量の拡張メモリを搭載しているとチェックに時間がかかりますから、省いてしまってもやむを得ないと思います。

EMM386.EXEの組み込み

EMM386.EXEは、80386以上のCPUがもっている**仮想8086モード**を利用した**仮想EMS**を有効にするためのデバイスドライバです。また、UMBメモリの割り当ても行います。EMM386.EXEは、XMSメモリからEMSメモリを割り当てるため、HIMEM.SYSよりもあとに組み込む必要があります。

EMM386.EXEにも、いくつかのオプションが用意されていますが、実際に使用するのはごく一部で済むはずです。基本的には、次の記述になります。

◉ EMM386.EXEの基本的な書式

```
DEVICE = A:¥DOS¥EMM386.EXE /UMB
```

/UMB というのは、UMB メモリを有効にするためのスイッチです。このスイッチを付けておかないと、DEVICEHIGH コマンドや INSTALLHIGH コマンドが使用できません。なお、EMM386.EXE で UMB メモリを割り当てるため、EMM386.EXE 自体を DEVICEHIGH コマンドで組み込むことはできないので注意してください。

UMB メモリを使用するなら、次の記述も忘れないでください。

```
DOS = UMB
```

/T スイッチは必要に応じて付ける

インストール時に作成された CONFIG.SYS では、/T = A:¥DOS¥EXTDSWAP.SYS というスイッチが付けられているはずです。

```
DEVICE=A:¥DOS¥EMM386.EXE /UMB /T=A:¥DOS¥EXTDSWAP.SYS
```

この/T スイッチは、DOS シェルにおいて**拡張タスクスワップ機能**を使用する際に用いるファイルを指定しています。

DOS シェルでは、複数のアプリケーションを同時に起動して、それぞれを切り替えながら使用することができます。これをタスクスワップと呼んでいます。しかし、DOS5 が出る以前に開発されたアプリケーションのなかには、タスクスワップを行うと、不具合が生じてしまうものがあるのです。そこで、それぞれのアプリケーションが動作していた環境を保存しておいて、切り替えた際に元どおりに環境を整える機能が加えられました。それが拡張タスクスワップ機能です。

このタスクスワップ機能を組み込むと、EMS メモリが 1 ページ分(16K バイト）使われてしまいますから、基本的には、/T スイッチは削除してしまうほうがよいでしょう。もしも、DOS シェルでタスクスワップを利用して不具合が発生するようであれば、改めて指定するようにしてください。

EMM386.EXEはEMSメモリのサイズ指定が不要

DOS5に付属のEMM386.EXEでは、XMSメモリ（EMBメモリ）とEMSメモリを区別して管理していたのに対して、DOS6では一括して管理するように仕様が変更になっています。プログラムの要求に応じて、自動的にXMSメモリまたはEMSメモリとして割り当ててくれるのです。そのため、DOS5で行っていたように、あらかじめEMSメモリのサイズを指定する必要はありません。

ただし、オプションとしては残されていますから、次のようにサイズを指定してもエラーにはなりません。

●/MスイッチによるEMSのサイズ指定（通常は不要）

```
DEVICE = A:¥DOS¥EMM386.EXE /M=2048 /UMB
```

/Mスイッチで指定可能なサイズは16〜32768（Kバイト）の範囲で、指定を省くと使用可能な全拡張メモリになります。ただし、EMSメモリの管理単位である16（Kバイト）に満たない部分は切り捨てられます。

同様の意味を持つ/Pスイッチもありますが、こちらはバイト数ではなく、16Kバイト単位のページ数で指定します。

●/PスイッチによるEMSのサイズ指定（通常は不要）

```
DEVICE = A:¥DOS¥EMM386.EXE /P=128 /UMB
```

「/P = 128」は、「/M = 2048」と同じ値を指定したことになります。

繰り返しますが、DOS6のEMM386.EXEでは、EMSメモリのサイズを指定する必要はありません。インストール時に自動的に作成されるCONFIG.SYSでは、以前の指定がそのまま引き継がれ、EMSメモリの指定があるかもしれません。その場合には、/Mスイッチまたは/Pスイッチの部分を削除しておきましょう。

Step 17

・日本語変換補助ドライバ：KKCSAV.SYS,KKCFUNC.SYS
・日本語変換ドライバ：NECAIK1.DRV,NECAIK2.DRV

日本語変換補助ドライバと日本語変換ドライバ(FEP)の組み込み

●ポイント●

・日本語変換用のデバイスドライバは、FEPと呼ばれる。
・FEPによっては、KKCSAV.SYSまたはKKCFUNC.SYSを組み込んでおく必要がある。
・NECAIK1.DRV,NECAIK2,DRVは、必ずKKCFUNC.SYSと共に用いる。

●解 説●

日本語入力に必要なデバイスドライバ

本来、パソコンは英数記号を表示する機能しか持っていないため、日本語の漢字やひらがななどを入力するためには、デバイスドライバを用いて機能を拡張してやる必要があります。それらのドライバは、**日本語フロントエンドプロセッサ**（Front-End Processor）と呼ばれています。フロントエンドというのはパソコンの手前側という意味で、ユーザーがキーボードから入力したデータに対して、前処理を行う機能を指しています。**日本語FEP**または、単に**FEP**と略して呼ばれるのが普通です。

DOS6にはFEPが付属していますが、たいていのユーザーは市販のFEPか、アプリケーションに付いてきたFEPを使用しています。機能的には、付属のFEPも市販のものに負けてはいないのですが、アプリケーションの対応度や、辞書ユーティリティなどの使いやすさから、市販のFEPをお勧めしたいと思います。

KKCSAV.SYSとKKCFUNC.SYS

FEPによっては、**日本語変換補助ドライバ**と呼ばれる**KKCSAV.SYS**または**KKCFUNC.SYS**をあらかじめ組み込んでおく必要があります。

KKCSAV.SYSやKKCFUNC.SYSは、DEVICEコマンドまたはDEVICEHIGHコマンドを使って組み込みます。

◉ KKCSAV.SYSとKKCFUNC.SYSの組み込み

```
DEVICE = A:¥DOS¥KKCSAV.SYS
DEVICEHIGH = A:¥DOS¥KKCSAV.SYS
```

```
DEVICE = A:¥DOS¥KKCFUNC.SYS
DEVICEHIGH = A:¥DOS¥KKCFUNC.SYS
```

KKCSAV.SYSは、DOS6に未対応のFEPを使用するための補助ドライバですが、基本的にDOS5で動作していたFEPは、そのままDOS6でも動作します。そのため、KKCSAV.SYSが必要になるのは、かなり古いバージョンのFEPに限られると考えられます。たとえば、『一太郎 Ver.4』に付属していたATOK7は、KKCSAV.SYSを用いるとよいでしょう。

◉ ATOK7とKKCSAV.SYSの組み込み

```
DEVICE(HIGH)=A:¥DOS¥KKCSAV.SYS
DEVICE(HIGH)=A:¥ATOK¥ATOK7A.SYS
DEVICE=A:¥ATOK¥ATOK7B.SYS
```

ただ、KKCSAV.SYSを使って無理にDOS6で利用するよりは、DOS6に付属のFEPを利用するか、またはバージョンアップのサービスを受けることを強くお勧めします。通常、KKCSAV.SYSは使うことはないため、削除してしまってもかまわないと考えてよいでしょう。

KKCFUNC.SYS は、DOS6（または DOS5）に対応している FEP を組み込む前に、必ず組み込んでおく補助ドライバです。たとえば、『一太郎 Ver.5』に付属の ATOK8 と組み合わせるなら、

```
DEVICE(HIGH)=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK¥ATOK8A.SYS
DEVICE(HIGH)=A:¥ATOK¥ATOK8B.SYS
```

となります。

KKCFUNC.SYS を組み込んであると、複数の FEP を組み込んでおいてから、それらを必要に応じて切り替えることが可能になります。ただ、FEP の常駐サイズが大きく、メモリがかなり減ってしまいますから、メリットよりもデメリットのほうが多いと思われます。なお、複数の FEP を組み込む場合、KKCFUNC.SYS は最初に 1 回だけ組み込むようにしてください。

付属 FEP の組み込み方

DOS6 に付属の FEP は、複数の文節を一度に漢字かな混じり文に変換できる連文節変換プログラムです。この FEP は、**NECAIK1.DRV** と **NECAIK2.DRV** という 2 つのドライバを同時に組み込んで使用します。必ず、NECAIK1.DRV を先に組み込むようにしてください。また、これらのドライバより前に、KKCFUNC.SYS を組み込んでおく必要があります。

```
DEVICE(HIGH)=A:¥DOS¥KKCFUNC.SYS
DEVICE(HIGH)=A:¥DOS¥NECAIK1.DRV (オプション)
DEVICE(HIGH)=A:¥DOS¥NECAIK2.DRV (オプション)
```

とりあえず必要なオプション類は、DOS6 のインストール時に付けられているはずです。

この FEP は、EMS メモリを利用して、常駐サイズが約 7K バイトと小さくなるようになっています。EMS メモリを使わなければ約 125K バイトと極端に大きくなってしまうので、必ず EMM386.EXE などで EMS を有効にしてから組み込むようにしてください。

辞書ファイルの保守ユーティリティとして **DICM** コマンドが、入力キー設定プログラムとして **NECAIKEY** コマンドが用意されています。

Step 18

特定のプログラムに指定したバージョン番号を返す【SETVER.EXE】

●ポイント●

・バージョンチェックでエラーになるプログラムは、SETVER.EXE で動作可能になることがある。

・SETVER コマンドには、あらかじめいくつかのプログラムが登録されている。

・SETVER.EXE はデバイスドライバとして組み込むほか、コマンドラインから実行してバージョンテーブルの設定を行う。

●解 説●

DOS のバージョンとプログラムのバージョンチェック

DOS の仕様は、**バージョン**が上がるたびに変わってきています。プログラムによっては、DOS の仕様変更によって何らかの不具合が起こることを避けるため、実行時に DOS のバージョンをチェックし、サポート外のバージョンであれば、エラーとして直ちに終了するものがあります。

しかし、バージョンが変わっていても、プログラムの実行には問題がないことも多いため、DOS5 からはプログラムに返すバージョン番号を自由に設定できる **SETVER.EXE** が用意されています。

SETVER.EXE を用いる際には、そのプログラムを実行しても本当に支障がないのかということをよく確認しておいてください。特に、DOS5 が出る前に開発されたディスク関連のプログラムでは、ディスクの内容が破壊されてしまうかもしれません。

起動時にバージョン違いによるエラーが起こらなければ、SETVER.EXE を組み込む必要はありません。

SETVER.EXE の組み込みとバージョンテーブルの設定

SETVER.EXE は、DEVICE コマンド、または DEVICEHIGH コマンドで組み込みます。オプションはありません。

◉ SETVER.EXE の組み込み

```
DEVICE = A:¥DOS¥SETVER.EXE
```

```
DEVICEHIGH = A:¥DOS¥SETVER.EXE
```

次に、システムが返すバージョン番号を指定するため、SETVER.EXE が持っている**バージョンテーブル**に登録する手続きを行います。コマンドラインから、SETVER コマンドを次の書式で実行してください。このとき、あらかじめ SETVER.EXE を組み込んであってもなくてもかまいません。

◉バージョンテーブルの設定書式

```
SETVER プログラム名 バージョン番号
```

プログラム名は、拡張子まで含めて指定します。ただし、ドライブ名やディレクトリ名は必要ありません。バージョン番号は、必ず X.XX のスタイルで指定します。

たとえば、SAMPLE.EXE というプログラムに 5.00 というバージョンを返すようにするには、

```
A:¥>SETVER SAMPLE.EXE 5.00
```

と実行します。

こうして設定したバージョンテーブルは、次回の起動時から有効になります。もちろん、CONFIG.SYS で SETVER.EXE を記述しておかなくてはなりません。

●登録状況の確認

SETVER.EXE のバージョンテーブルの登録状況を確認するには、オプションなしで実行するか、SETVER.EXE のあるパス名を指定して実行します。

◉バージョンテーブルの登録状況を確認する

```
A:¥>SETVER
```

```
A:¥>SETVER A:¥DOS
```

まだ何も登録していなくても、ためしに SETVER コマンドを実行してみてください。すでに、いくつかのプログラムが登録されているはずです。リダイレクトを使って出力結果をファイルにしておくと、コマンドを誤って使用した場合の修正に役立ちます。

◉バージョンテーブルの内容を VERTBL.DAT に出力する

```
A:¥>SETVER >VERTBL.DAT
```

◉ SETVER コマンドの実行例

```
A:¥>SETVER

EXE2BIN.EXE     5.00
NETX.COM        5.00
NET5.COM        5.00
EMSNETX.EXE     5.00
XMSNETX.EXE     5.00
DRS.COM         3.00
DINSTAL2.EXE    3.30
DAL2MGR.COM     3.00
DINSTBC.EXE     3.30
FORMATA.EXE     5.00

注意: SETVER デバイスは読み込まれていません. SETVER のバージョン通知を有効に
      するには CONFIG.SYS 中で SETVER.EXE デバイスを読み込まなければなりません.
```

※使用している DOS の出荷時期によっては、内容が異なっているかもしれません。

バージョンテーブルの変更と削除

SETVER.EXEで設定したバージョンテーブルを変更するには、新たなバージョンを指定して登録するだけです。たとえば、5.00に設定済みのSAMPLE.EXEを6.00に変更するなら、次のように実行してください。

```
A:¥>SETVER SAMPLE.EXE 6.00
```

変更ではなく、バージョンテーブルから削除してしまうには、プログラム名と/Dまたは/DELETEスイッチを付けて実行します。その際、/Qまたは/QUIETスイッチを一緒に付けると、削除メッセージを表示しません。

◉プログラム名をバージョンテーブルから削除する

```
A:¥>SETVER SAMPLE.EXE /D
```

```
A:¥>SETVER SAMPLE.EXE /D /Q
```

変更や削除を行った場合、SETVERコマンドを実行して表示されるバージョンテーブルには直ちに反映されますが、実際には有効になっていません。次回の起動時から有効となります。

Step 19

プリンタドライバの組み込み【PRINT.SYS】

●ポイント●

- MS-DOSに付属のPRINT.SYSは、PC-PR201系プリンタに対応している。
- アプリケーションでは専用のプリンタドライバを使用するので、組み込む必要は特にない。

●解 説●

DOS6のプリンタドライバ

DOS6に付属のプリンタドライバ**PRINT.SYS**は、デバイスドライバとしてCONFIG.SYSで組み込むタイプです。対応しているのは、NEC製のPC-PR201系のプリンタですが、ほかのメーカーのプリンタでも、PC-PR201との互換モードを備えていれば利用することができます。

DOSに付属のプリンタドライバは、主にDOSコマンドの実行結果をプリンタに出力するためのものです。アプリケーションは、たいてい専用のプリンタドライバが自動的に組み込まれるようになっていますから、このプリンタドライバを組み込んでおく必要はありません。したがって、必要がない限りは使わないようにするほうがよいでしょう。

◉ PRINT.SYSの組み込み

```
DEVICE = A:¥DOS¥PRINT.SYS
```

```
DEVICEHIGH = A:¥DOS¥PRINT.SYS
```

PRINT.SYS のオプションには、次のものがあります。

◉ PRINT.SYS のオプション

/U	ユーザー定義文字（外字）を印字する場合に指定する。 ただし、ユーザー定義文字を印字するには、起動ドライブのルートディレクトリにユーザー定義文字ファイル（USKCG24.SYS）を置き、MS-DOS の起動前にプリンタの電源をオンにしておく必要がある。 印字可能な文字は、JIS コードの 7621～7674H まで（84 文字）。
/P1	2 バイトコード文字(全角文字)を 1 バイトコード文字(半角文字)の 1.5 文字分の幅で印字する。 このスイッチを指定しなければ、2 バイトコード文字は 1 バイトコード文字の 2 文字分の幅で印字される。
/F	PC-H98 のノーマルモードで、プリンタをフルセントロニクス仕様で制御できるようにする（PC-H98 のみで指定可能）。

通常の使用では、オプションを付ける必要はありません。自分で作成した文字をユーザー定義文字として印字する場合には、/U スイッチを付けてください。USKCG24.SYS は、USKCGM コマンドで作成できます。

たとえば、PRINT.SYS を組み込んだ状態で COPY キーを押すと、画面の表示内容が印字されます。文字の属性も出力するなら、GRPH ＋ COPY キーを押してください。ただし、一般のアプリケーションでは、たいてい動作しないようになっています。利用できるのは、DOS のコマンドを実行した状態のみと考えるほうがよいでしょう。

なお、プリンタには 16 ドット系と 24 ドット系があり、初期設定は 24 ドット系になっています。16 ドット系に変更するには、SWITCH コマンドを実行してください。

Step 20

RS-232Cインターフェイスドライバの組み込み【RSDRV.SYS】

●ポイント●

- RSDRV.SYSは、RS-232Cインターフェイスを用いた通信を行うときなどに組み込むドライバ。
- 現在では、RS-232C用のドライバはアプリケーション側が持っているため、RSDRV.SYSを組み込む必要はなくなっている。

●解 説●

RS-232Cインターフェイスとは

PC-9800シリーズの本体背面には、**RS-232Cインターフェイス**の差し込み口が付いています。**RSDRV.SYS**は、このRS-232Cインターフェイスを用いた通信を行う場合などに組み込むドライバです。オプション類はありません。

```
DEVICE = A:¥DOS¥RSDRV.SYS
```

```
DEVICEHIGH = A:¥DOS¥RSDRV.SYS
```

パソコン通信で使うモデムをRS-232Cケーブルで接続した場合は、パソコン通信ソフトがRS-232Cドライバを内蔵しているため、RSDRV.SYSは必要ありません。また、画像を読み取るイメージスキャナの場合も、たいていは画像ソフトが機種に合わせたドライバを用意しています。PRINT.SYSと同様、RSDRV.SYSを組み込む必要はほとんどなくなっているのです。MAXLINK-LITEを利用する際も、**MAXLINK.DRV**を組み込むだけで、RSDRV.SYSは組み込まないでください。

Step 21

マウスドライバの組み込み【MOUSE.SYS】

●ポイント●

・MOUSE.SYS は、アプリケーションやゲームソフトでマウスを使うためのドライバ。
・Windows や多くのアプリケーションでは、独自のマウスドライバを持っているため、MOUSE.SYS は必要ない。
・DOS シェル、DBLSPACE コマンド、DEFRAG コマンド、MSD コマンド、QBACKUP コマンド、SCANDISK コマンドなど、DOS に付属のユーティリティでマウスを利用する際には、MOUSE.COM のほうを利用する。

●解 説●

マウスドライバ MOUSE.SYS

入力装置としての**マウス**は、今やキーボードと共に広く使われるようになっています。**MOUSE.SYS** は、CONFIG.SYS で組み込むマウス用のドライバです。

◉マウスドライバの組み込み

```
DEVICE = A:¥DOS¥MOUSE.SYS
```

```
DEVICEHIGH = A:¥DOS¥MOUSE.SYS
```

このMOUSE.SYSについても、必要になるケースというのは、たいへん少なくなってきていると言えるでしょう。たいていのアプリケーションやゲームソフトでは、独自のマウスドライバを持っているからです。

マウスが利用できるアプリケーションやゲームソフトにおいて、マニュアルで「MOUSE.SYSが必要」とある場合にのみ、組み込むようにしてください。Windows環境では、このMOUSE.SYSは必要ありません。

MOUSE.SYSとMOUSE.COMの使い分けに注意

DOS6には、MOUSE.SYSのほかに常駐プログラムタイプの**MOUSE.COM**というマウスドライバが付属になっています。このMOUSE.COMは、DOSシェル、DBLSPACEコマンド、DEFRAGコマンド、MSDコマンド、QBACKUPコマンド、SCANDISKコマンドなど、DOSに付属のユーティリティでマウスを利用する専用のドライバですから、MOUSE.SYSとは区別するようにしてください。MOUSE.SYSを組み込んであっても、DOS付属のユーティリティ類ではマウスが機能しませんし、MOUSE.COMを実行してあっても、一般のアプリケーションやゲームソフトではマウスを利用できないのです。

なお、MOUSE.SYSとMOUSE.COMは同時に組み込むことになっても、不具合は発生しません。

ベクタ番号の指定方法

何かのプログラムを実行している最中に、その処理を中断して別のプログラムを動作させることを、**割り込み（割り込み処理）**と呼んでいます（割り込みについては、「Part3 応用編」で詳しく解説しています）。

MS-DOSで割り込みを行うためには、割り込み処理の先頭アドレスがセットされた**割り込みベクタ**を呼び出すようになっており、MS-DOSが基本としている8086というCPUでは、00～FFHまでの256種類の割り込みベクタが用意されています。

多くの常駐プログラムやデバイスドライバは、割り込み処理を伴うため、それぞれが割り込みベクタを確保しています。このことを割り込みベクタを**フック**するといいます。

ほかのプログラムがすでにフックしている割り込みベクタを使ってしまわないように、たいていのプログラムでは、あとから同じ割り込みベクタを利用するプログラムを組み込もうとすると、エラーになるようになっています。しかし、なかにはエラーとならず、重複してしまうこともあるのです。その際には、プログラムの動作に何らかの不具合が生じてしまいます。常駐プログラムやデバイスドライバを組み込んでいて、正常に動作しない場合には、割り込みベクタの重複が発生していないかチェックしてみましょう。

MOUSE.SYS では、初期設定で割り込みベクタ 15H を利用するようになっていますが、ほかのプログラムと重複してしまう場合には、/I スイッチを用いて、11H、12H、14H のいずれかに変更することが可能です。たとえば、11H に変更するなら、

```
DEVICE = A:¥DOS¥MOUSE.SYS /I:11
```

のように設定してください。ただし、あまり変更する必要はないはずです。

なお、ハイレゾリューションモードでは、割り込みベクタは 0EH に固定されています。

Step 22

RAMディスクドライバの組み込み【RAMDISK.SYS】

●ポイント●

・RAMディスクを作成するにはRAMDISK.SYSを組み込む。
・作成できるRAMディスクは最大で14.5Mバイト。

●解　説●

RAMディスクドライバの組み込み方

RAMディスクは、拡張メモリの一部をディスク装置のように扱うことができるようにしたものです。ヘッドの移動時間などがないため、最も高速に読み書きが可能なドライブとしてたいへん重宝します。

RAMディスクは、**RAMDISK.SYS**というデバイスドライバで設定を行います。

◉ RAMDISK.SYSの書式

DEVICE = A:¥DOS¥RAMDISK.SYS ディスク容量（オプション）
DEVICEHIGH = A:¥DOS¥RAMDISK.SYS ディスク容量（オプション）

ディスク容量は、128Kバイト単位で128〜14848の範囲から指定できます(既定値は128)。ただし、搭載されているメモリによっては、最大容量が小さくなることもあります。

オプションの内容

オプションには、次のものがあります。

論理セクタ長	RAM ディスクの論理セクタ長を 512(バイト)または 1024(バイト)で指定する（既定値は 1024)。
ディレクトリ数	RAM ディスクのディレクトリ数を 128～1024 の範囲で指定する。 省略した場合には、RAM ディスクの容量によって、次表のように決められる。
/M または/E	/M を指定すると、基本メモリ内に RAM ディスクを作成する。 /E を指定すると、EMS メモリ内に RAM ディスクを作成する。 これらのスイッチを省略すると、EMB メモリ内に RAM ディスクを作成する。

●論理セクタ長

論理セクタ長とは、ディスクを管理する際の最小単位のことで、小さいほうが効率よく記憶できますが、不連続状態で記憶されていると、アクセス回数が増えて速度が低下してしまいます。ただし、もともと高速な RAM ディスクでは、どちらを指定しても大差はないでしょう。

●ディレクトリ数

ディレクトリ数とは、ディスクのルートディレクトリで管理可能なファイルとディレクトリの数のことで、大量のファイルを扱う場合などには、この数を増やしておく必要があります。余分に設定すると、それだけディスク容量が減ってしまうので、たいていは初期値のままでよいでしょう。

論理セクタ長を指定する場合には、必ず容量も一緒に指定してください。同様に、ディレクトリ数を指定するには、容量と論理セクタ長も同時に指定します。

◉ディレクトリ数省略時の既定値

RAM ディスク容量	ディレクトリ数	RAM ディスク容量	ディレクトリ数
128K バイト〜1MB バイト未満	128	8〜9M バイト未満	640
1〜2M バイト未満	192	9〜10M バイト未満	704
2〜3M バイト未満	256	10〜11M バイト未満	768
3〜4M バイト未満	320	11〜12M バイト未満	832
4〜5M バイト未満	384	12〜13M バイト未満	896
5〜6M バイト未満	448	13〜14M バイト未満	960
6〜7M バイト未満	512	14〜14.5M バイト未満	1024
7〜8M バイト未満	576		

◉ディスク容量、論理セクタ長、ディレクトリ数の設定例

```
DEVICEHIGH = A:¥DOS¥RAMDISK.SYS 3072 512
DEVICEHIGH = A:¥DOS¥RAMDISK.SYS 2048 1024 384
```

●/M スイッチと/E スイッチ

/M スイッチは、何より貴重な基本メモリを使うという指定なので、使われることはないと言ってよいでしょう。

/E スイッチを使うと、EMS メモリから割り当てられますが、その場合には、16K バイト単位でバイト数を指定できます。ただし、EMB メモリから割り当てるほうが若干高速になるので、通常はこれらのスイッチは付けないでおきます。

Step 23

一部機種用ドライバの組み込み【BCKWHEAT.SYS】

●ポイント●

・機種によっては、インストール時にカレンダ時計の不具合に対処するための BCKWHEAT.SYS が自動的に組み込まれる。

●解 説●

BCKWHEAT.SYS の役割

DOS6 の補足マニュアルにもありますが、PC-9821Ts/Ap2/As2/Bp/Bs/Be、PC-9801BA2/BS2/BX2(ほかに、ノートパソコンの一部機種も含まれます)では、インストールを行うと CONFIG.SYS に次の 1 行が加えられます。

```
DEVICE=A:¥DOS¥BCKWHEAT.SYS
```

BCKWHEAT.SYS というのは、これらの機種で確認されている"本体内のカレンダ時計が年越しをしない"という不具合に対処するためのドライバです。常駐サイズはごくわずかなので、もしも組み込まれていたら、そのまま使うようにするほうがよいでしょう。

該当しない機種では、もちろんまったく必要のないドライバです。DOS をインストールしたディレクトリにもコピーされていないので安心してください。

Step 24

AUTOEXEC.BATの基本コマンド①【ECHO,@】

●ポイント●

・AUTOEXEC.BATの最初には「@ECHO OFF」を記述しておくと、余計なメッセージが表示されない。
・ECHOコマンドを使って、バッチでユーザーの指定したメッセージを表示させることができる。
・空行だけを表示するには、ECHOの直後に「.」(ピリオド)を記述する。

●解 説●

画面表示をコントロールするECHOコマンドと@コマンド

ECHOコマンドは、DOSコマンドとしてコマンドラインから実行することも可能ですが、たいていはバッチファイルで用いられます。主な用途としては、

- **コマンドのエコーバックを停止/再開する**
- **ユーザーが指定したメッセージを表示する**

という2通りが考えられます。

コマンドの**エコーバック**というのは、バッチファイル内で記述したコマンド文字列の表示ということです。次のような、簡単なバッチファイルで試してみましょう。

◉ TEST1.BAT の内容

```
DIR /W
ECHO
```

◉ TEST1.BAT の実行結果

```
E:¥>TEST1

E:¥>DIR /W

 ドライブ E: のボリュームラベルは MS-RAMDRIVE
 ディレクトリは E:¥

TEST1.BAT       TEST2.BAT       TEST3.BAT       TEST4.BAT       TEST5.BAT
        5 個                344 バイトのファイルがあります
                      6,250,496 バイトの空きがあります

E:¥>ECHO
ECHO は ON

E:¥>
```

結果は、コマンドラインからキーボード入力して実行した場合と同じになります。ECHO コマンドをオプションなしで実行すると、現在のエコーバックの設定状況を表示します。

●エコーバックの停止

次に、エコーバックを停止するために、ECHO OFF を記述してみます。

◉ TEST2.BAT の内容

```
ECHO OFF
DIR /W
ECHO
```

◉ TEST2.BAT の実行結果

```
E:¥>TEST2

E:¥>ECHO OFF

 ドライブ E: のボリュームラベルは MS-RAMDRIVE
 ディレクトリは E:¥

TEST1.BAT        TEST2.BAT        TEST3.BAT        TEST4.BAT        TEST5.BAT
        5 個                  344 バイトのファイルがあります
                        6,250,496 バイトの空きがあります
ECHO は OFF
E:¥>
```

このように、エコーバックを停止してしまうと、そのあとのプロンプトとコマンド文字列が表示されず、コマンドの実行結果だけが表示されるようになります。結果を見やすくするため、バッチファイルの最初ではエコーバックを停止しておくほうがよいでしょう。

●「@」記号の意味

次に、ECHO コマンドの前に@という記号を付けてみます。

◉ TEST3.BAT の内容

```
@ECHO OFF
DIR /W
ECHO
```

◉ TEST3.BAT の実行結果

```
E:¥>TEST3

 ドライブ E: のボリュームラベルは MS-RAMDRIVE
 ディレクトリは E:¥

                                  TEST3.BAT        TEST4.BAT        TEST5.BAT
                              344 バイトのファイルがあります
                        6,250,496 バイトの空きがあります
```

結果は、最初のECHOコマンドからエコーバックが停止されます。このように、「@」(アットマーク)は、直後に記述された1コマンドだけについてエコーバックを停止するコマンドなのです。

ECHOコマンドで停止したエコーバックは、やはりECHOコマンドで再開することができます。

◉ TEST4.BATの内容

```
@ECHO OFF
DIR /W
ECHO
ECHO ON
ECHO
```

◉ TEST4.BATの実行結果

```
E:¥>TEST4

 ドライブ E: のボリュームラベルは MS-RAMDRIVE
 ディレクトリは E:¥

TEST1.BAT        TEST2.BAT        TEST3.BAT        TEST4.BAT        TEST5.BAT
        5 個                     344 バイトのファイルがあります
                           6,250,496 バイトの空きがあります
ECHO は OFF

E:¥>ECHO
ECHO は ON

E:¥>
```

ECHO ONを実行すると、その次のコマンドからエコーバックが行われます。

なお、バッチファイルが終了すると、エコーバックはバッチファイルの実行前の設定に戻ります。

ECHO コマンドによるメッセージの表示

ECHO コマンドは、バッチファイルの実行中にメッセージを表示させるためにも用いられます。

◉ TEST5.BAT の内容

```
@ECHO --- エコーバックを停止します ---
@ECHO OFF
ECHO --- DIRコマンドを実行します ---
DIR /W
ECHO --- ECHOコマンドを実行します ---
ECHO
ECHO --- エコーバックを開始します ---
ECHO ON
@ECHO --- ECHOコマンドを実行します ---
ECHO
```

◉ TEST5.BAT の実行結果

```
E:¥>TEST5
--- エコーバックを停止します ---
--- DIRコマンドを実行します ---

 ドライブ E: のボリュームラベルは MS-RAMDRIVE
 ディレクトリは E:¥

TEST1.BAT       TEST2.BAT       TEST3.BAT       TEST4.BAT       TEST5.BAT
        5 個            344 バイトのファイルがあります
                  6,250,496 バイトの空きがあります
--- ECHOコマンドを実行します ---
ECHO は OFF
--- エコーバックを開始します ---
--- ECHOコマンドを実行します ---

E:¥>ECHO
ECHO は ON

E:¥>
```

ECHO コマンドでメッセージを表示する際には、必ずエコーバックを停止しておくか、@コマンドを付けておくようにしましょう。

続けてメッセージを表示させる場合、間に空行を入れるには、ECHO コマンドの直後にピリオド(.)だけを記述します。ECHO コマンドとピリオドの間にスペースを入れないでください。

Step 25

AUTOEXEC.BAT の基本コマンド②【SET】

●ポイント●

- 環境変数は文字列の代入を行う。
- 環境変数は SET コマンドで設定する。
- SET コマンドでは、=（イコール）の前後にスペースを入れない。
- 環境変数を格納する環境エリアは SHELL コマンドの/E スイッチで拡張できる。
- 環境変数 COMSPEC と TEMP はシステムが使用する。

●解　説●

プログラミングで用いられる変数とは

パソコンに 1+2=3 という計算をさせるには、次のように指示することになります。

```
A = 1
B = 2
C = A + B
```

この場合の「=」は「等しい」ではなく、「右辺を左辺に代入する」という意味です。左辺にある A、B、C という文字は、すべて計算のために確保されたメモリを示していると考えてください。

つまり、これらの式は、Aというメモリに値「1」を代入して、次にBというメモリに値「2」を代入、最後にAとBを加えて、Cというメモリに代入する……という手順になるのです。最後の式の右辺にあるAとBは、メモリではなく、それぞれのメモリに入っている値を示しています。

これらの式で用いられているA〜Cの記号を、**変数**と呼んでいます。変数は、何かを入れるために用意された箱のようなもので、中身は自由に入れ替えることができます。

変数を用いるメリットとして、具体的に値を決められないようなデータがあっても、プログラミングが可能になるということがあります。たとえば、必要なファイルを作成するドライブ名を、「A:」や「B:」のような指定ではなく、DRIVEのような変数を用いてプログラミングしておきます。そして、ユーザーが実行する際に、変数DRIVEに好きなドライブ名を代入してもらうのです。もし、具体的なドライブ名を用いてしまえば、融通のきかないプログラムになってしまうでしょう。

環境変数とSETコマンド

プログラム中で、ユーザーの環境に合わせて指定してもらうように確保された変数のことを、特に**環境変数**と呼びます。そして、ユーザーが環境変数に代入するためのコマンドが、**SET**コマンドです。

たとえば、DRIVEという環境変数に「C:」という文字列を代入するには、

```
SET DRIVE=C:
```

のようにします。

この場合、「=」の前後にスペースを入れないように注意してください。

次のように記述すると、「DRIVE＋スペース」という環境変数に、「スペース＋C:」という文字列を代入する指定になってしまうのです。

```
SET DRIVE = C:
```

たいていのプログラムでは、余分なスペースは削除してくれるはずですが、バッチファイルで環境変数を利用する場合には、そのような処理をしてくれません。

環境エリアの拡張

SET コマンドで設定される環境変数は、**環境エリア**と呼ばれるメモリに格納されます。この環境エリアは、初期設定で256バイトになっていますが、環境変数が増えて領域が足りなくなった場合には、CONFIG.SYS内のSHELLコマンドで/Eスイッチを指定してください。

たとえば、512バイトにするなら、

```
SHELL = A:¥DOS¥COMMAND.COM A:¥DOS /P /E:512
```

のように記述します。

詳しくは、SHELLコマンドの解説を参照してください。

設定しておくとよい環境変数

DOSシステムでは、あらかじめいくつかの環境変数が用いられています。それらのうち、**COMSPEC** と **TEMP**（または **TMP**）は、ぜひ覚えておいてください。

◉ COMSPEC と TEMP の設定例（AUTOEXEC.BAT）

```
SET COMSPEC=A:¥DOS¥COMMAND.COM
SET TEMP=F:¥
```

これらは、インストール時に自動的に設定されていることもあると思います。**COMSPEC** というのは、CONFIG.SYSのSHELLコマンドで指定するコマンドインタプリタの再読み込み先の情報です。もし、SHELLコマンドで指定していなければ、必ず設定しておいてください。なお、コマンドインタプリタについては、SHELLコマンドの解説を参照してください。

環境変数 TEMP（TMP）では、プログラムが作業を行うために一時的に使用するファイルの作成場所を指定します。作業ファイルの処理が高速になるほど、プログラムの実行速度も向上しますから、最も高速なドライブに指定しておくとよいでしょう。

ただし、ハードディスクはあまりお勧めできません。かなりディスクを酷使することになるからです。TEMP には、ぜひとも RAM ディスクのドライブを指定するようにしてください。

DOS のシステムが用いている環境変数には、次のものがあります。

◉ DOS システムが使う環境変数

環境変数	内　容
COMSPEC	COMMAND.COM を再読み込みするためのパス名
DOSDIR	DOS シェルの起動に必要なファイルがあるパス名
PATH	外部コマンドの検索パス名
PROMPT	プロンプトの形や色を指定するための文字列
TEMP/TMP	作業用ファイルの作成場所（ドライブ名およびディレクトリ名）
COPYCMD	COPY、MOVE、XCOPY に指定するオプションの文字列
DIRCMD	DIR コマンドに指定するオプションの文字列

Step 26

AUTOEXEC.BATの基本コマンド③【PATH】

●ポイント●

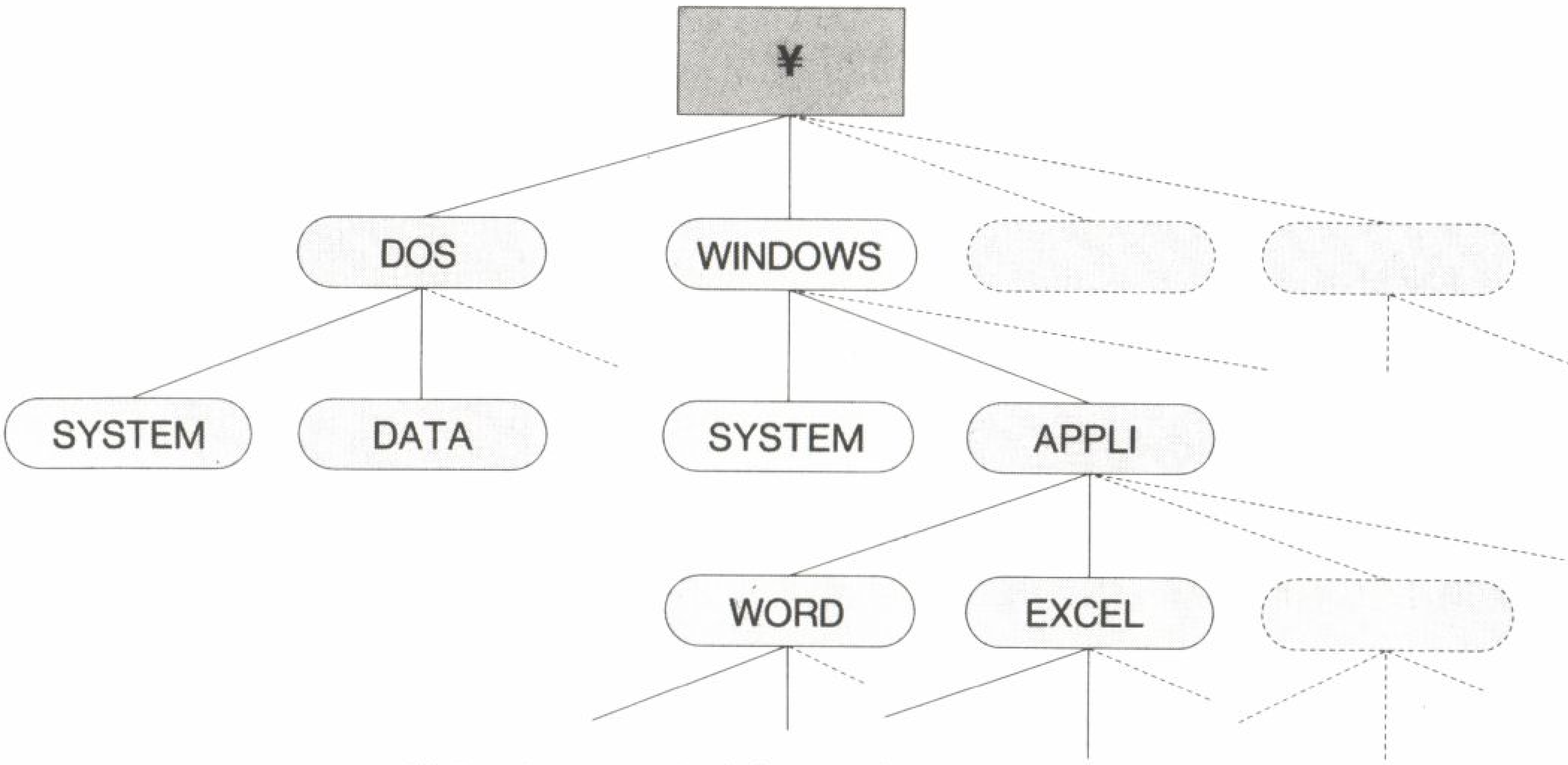
・よく利用する実行ファイルがあるディレクトリには、PATHコマンドでパスを通しておく。
・PATHは全部で122バイトまでしか有効にならない。
・PATHは先に記述したところほど優先される。
・PATHは環境変数の1つなので、SETコマンドでも設定できるが、PATHコマンドを用いるほうが長いパスを設定できる。

●解　説●

「パスを通す」という意味

DOSでは、多数のファイルを効率よく管理するため、ファイルの入れ物としてのディレクトリを階層化するしくみ、つまり**階層ディレクトリ**が用いられています。大元のディレクトリ（**ルートディレクトリ**）から枝分かれしているようにも考えられるため、**ツリー構造**とも呼ばれています。

◉階層ディレクトリ（ツリー構造）の様子



この階層ディレクトリでは、コマンドラインから直ちに実行可能なプログラムは、カレントドライブのルートディレクトリにあるファイルに限られています。「**パスを通す**」というのは、それ以外のドライブやディレクトリにある実行ファイルであっても、直ちに実行できるように、バイパスを通してやることと考えればよいでしょう。

すべての場所にパスを通すことはできませんから、どこからでも実行したいファイルのあるところに対して設定しておきます。

PATH コマンドの設定

パスを通すためには、PATH コマンドを実行しますが、通常は AUTOEX-EC.BAT 内で用います。

◉ PATH コマンドの書式

```
PATH パス名;パス名;パス名;……
```

◉ PATH コマンドの例

```
PATH A:¥DOS;A:¥DOS¥BATCH;A:¥TOOL;A:¥EDITOR
```

設定するパスとパスの間は、セミコロン（；）で区切ります。なお、コマンドラインからセミコロンだけを付けて実行すると、すべてのパスの設定が解除されます。

◉すべてのパス設定を解除する

```
A:¥>PATH;
```

● PATH コマンドの長さ

PATH コマンドで認識されるのは、セミコロンも含めて全部で 122 バイト（半角 1 文字が 1 バイト）までですから、あまり深い階層のパス名を指定していると、すぐに足りなくなってしまいます。ディレクトリの作成に工夫が必要でしょう。

●ファイルの検索順序

ユーザーがコマンドラインから実行したファイルは、まずカレントディレクトリから検索され、次に、PATH コマンドで設定してあるパス名を先頭から順番に検索していきます。このことは、次の 2 つの結果をもたらします。

- **先に記述してあるほど、実行されるまでの時間が短くなる。**
- **同名ファイルであれば、先に設定してあるものが起動される。**

速度の点は、体感できるほどの違いにはならないでしょうから、神経質になることはありません。考慮すべきは、まったく同じ名前の実行ファイルがある場合には、先に設定してあるものが起動されるという点でしょう。同名ファイルがあれば、片方を別の名前に変更してしまうか、変更が許されなければ、バッチファイルを使って起動するようにしておきましょう。

実行ファイルには COM、EXE、BAT という拡張子が付いていますが、ファイル名が同じであれば、COM ファイルの優先順位が最も高く、以降、次の順に実行されます。

```
COM → EXE → BAT
```

PATHは環境変数

PATHコマンドで設定したパス名は、PATHという環境変数として環境エリアに保管されます。そのため、SETコマンドを用いてパスを通すことも可能です。

```
SET PATH =パス名;パス名;パス名……
```

ただしSETコマンドの場合、全体で118バイトのパスしか設定できません。少しでも長いパスを設定したければ、PATHコマンドを使ってください。

制限以上のパスを通す方法

前述したように、PATHコマンドで設定できるパスは、全体で122バイト以内です。パスを通したいディレクトリが多くてはみ出てしまった場合には、これから紹介するテクニックを試してみてください。

まず、CONFIG.SYSのSHELLコマンドで、環境エリアを512バイト程度まで拡張しておきます。次に、AUTOEXEC.BAT内で、環境変数PATHと同じ内容の環境変数を設定しておきます。バッチファイル内で、環境変数名を%で囲って記述すると、その環境変数の内容に置き換えられます（この場合は%PATH%）。

◉ダミーの環境変数DEF_PATHにPATHと同じ内容を設定する

```
PATH A:¥DOS;A:¥DOS¥BATCH;A:¥TOOL;A:¥EDITOR
SET DEF_PATH=%PATH%
```

こうしておけば、アプリケーションを起動するバッチファイルで、簡単にPATHの設定を切り替えることができます。

◉ PATHを切り替えるバッチファイルの例

```
PATH <新たなパスの設定>
      (アプリケーションの起動コマンド)
SET PATH =%DEF_PATH%
```

まず、新たなPATHの設定を行い、それからアプリケーションを起動して、最後に、PATHの内容を元に戻しています。

Step 27

AUTOEXEC.BAT の基本コマンド④【PROMPT】

●ポイント●

・PROMPT コマンドでは、プロンプトの形状を設定する。

●解 説●

DOS プロンプトのカスタマイズ

DOS システムで、コマンドの入力位置に表示される**プロンプト**は、ユーザーによるカスタマイズが可能になっています。インストールした直後のAUTOEXEC.BAT には、**PROMPT** コマンドで次のように設定されているはずです。

```
PROMPT $P$G
```

これにより、プロンプトにはカレントドライブとカレントディレクトリ名が表示されます。

```
A:¥>
```

この設定が最もよさそうです。

「$P」や「$G」など、PROMPT コマンドで用いられる記号には、次の意味があります。

◉ PROMPT コマンドのオプション

文字列	意　味
$Q	「=」の文字
$$	「$」の文字
$T	現在時刻
$D	現在日時
$P	カレントドライブとカレントディレクトリ
$V	DOS のバージョン番号
$N	カレントドライブ
$G	「>」の文字
$L	「<」の文字
$B	「|」の文字
$	復帰／改行
$E	エスケープコード
$H	バックスペース

いくつか、設定例を挙げておきましょう。

◉ PROMPT コマンドの設定例

設定	プロンプト	内容
PROMPT THHH$G	11:20:45>	現在時刻
PROMPT $E[36m$PGE[m	A:¥>	カレントドライブとカレントディレクトリ（水色で表示）
PROMPT コマンド→	コマンド→	文字列「コマンド→」

エスケープシーケンスを使うとプロンプトに色を付けることもできますが、その際には、最後に「$E[m」を記述して、通常の色に戻すようにしてください。

プロンプトの初期設定は、カレントドライブのあとに「>」が表示される状態です。つまり、次の設定と同じです。

```
PROMPT $N$G
```

初期設定のプロンプトに戻すには、オプションなしで実行してください。

```
A:¥>PROMPT
```

Step 28

AUTOEXEC.BAT の基本コマンド⑤【LOADHIGH(LH)】

●ポイント●

・常駐プログラムを UMB メモリに常駐させるには LOADHIGH コマンドまたは LH コマンドを用いる。

●解 説●

常駐プログラムの UMB メモリへのロード

常駐プログラムを UMB メモリにロード（読み込み）するには、CONFIG.SYS 内で INSTALLHIGH コマンドを用いる方法があります。しかし、INSTALLHIGH コマンドでは組み込むことのできない常駐プログラムもあるため、基本的に、常駐プログラムを UMB メモリに組み込むには、AUTOEXEC.BAT 内で **LOADHIGH** コマンド、略して **LH** コマンドを用いてください。

◉ LOADHIGH（LH）コマンドの書式

LOADHIGH 常駐プログラム名 （常駐プログラムのオプション）
LH 常駐プログラム名 （常駐プログラムのオプション）

たとえば、DOSKEY を UMB メモリに常駐させるなら次のようにします。

```
LH DOSKEY
```

デバイスドライバを DEVICEHIGH コマンドで組み込む場合のように、LH コマンドで UMB メモリに常駐させる際には、常駐時のサイズよりも大きな UMB メモリが必要になります。必要なメモリが不足していると、基本メモリに常駐します。

Step 29

AUTOEXEC.BAT の基本コマンド⑥【REM】

●ポイント●

・行の先頭に REM を記述すると、その行はコメント行として無視される。
・AUTOEXEC.BAT では；（セミコロン）は使えない。

●解 説●

コメント行は無視される

AUTOEXEC.BAT のなかで、先頭に REM という文字列があると、その行は無視されるようになっています。わかりやすくするために途中でコメント（解説）を入れたり、同じコマンドで複数の設定を使い分けたりする際に利用するとよいでしょう。

◉コメントの例

```
REM ここでWindowsを起動する
```

REM コマンドのあとには、必ず半角スペースを入れるようにしてください。

なお、CONFIG.SYS では、REM コマンドの代わりに；(セミコロン) も使えますが、AUTOEXEC.BAT 内では使えないので注意が必要です。

Step 30

DOS コマンド専用マウスドライバの組み込み【MOUSE.COM】

●ポイント●

・DOS シェル、DBLSPACE コマンド、DEFRAG コマンド、MSD コマンド、QBACKUP コマンド、SCANDISK コマンドなど、DOS に付属のユーティリティでマウスを利用する際には MOUSE.COM を組み込む。

●解 説●

DOS コマンド専用マウスドライバの組み込みと常駐の解除

DOS シェル、DBLSPACE コマンド、DEFRAG コマンド、MSD コマンド、QBACKUP コマンド、SCANDISK コマンドなど、DOS に付属のユーティリティで**マウス**を利用する際には、常駐型のマウスドライバ **MOUSE.COM** を実行してください。

MOUSE.COM は、一般のゲームソフトやアプリケーションでマウスを利用するためのドライバではありません。そうしたソフトウェアでは、MOUSE.SYS を組み込むようにしてください（ただし、不要なソフトウェアもあります）。

MOUSE.COM は、次のようにコマンドラインから実行するか、バッチファイル内に記述しておきます。

◉ MOUSE.COM の組み込み

```
A:¥>MOUSE
```

MOUSE.COMの常駐を解除するには、/Rスイッチを付けて実行してください。

◉ MOUSE.COMの常駐解除

```
A:¥>MOUSE /R
```

MOUSE.COMには、次のスイッチがありますが、ほとんど付加する必要はありません。あくまで参考と考えてください。

◉ MOUSE.COMのオプション

スイッチ	意　味
/Sn	水平、垂直方向の感度（n = 0〜100）
/Hn	水平方向の感度（n = 0〜100）
/Vn	垂直方向の感度（n = 0〜100）
/Nn	緩慢動作の遅延値（n = 0〜255）
/Pn	加速曲線番号の指定（n = 1〜4）
/In	割り込みベクタの指定（n = 3〜6）
/Mn	カーソルマスク設定の許可（n = 0:許可、1:禁止）

インストール時に作成されるAUTOEXEC.BATの最後には、次の記述があります。

```
MOUSE
DOSSHELL
MOUSE /R
```

これは、DOSシェルを起動する前にMOUSE.COMを常駐させ、DOSシェルの終了時にMOUSE.COMの常駐を解除するという内容になります。したがって、DOSシェルを利用しないなら、3行まとめて削除してください。

Step 31

ディスクキャッシュの組み込み【SMARTDRV.EXE】

●ポイント●

・ディスクキャッシュには読み込みキャッシュと書き込みキャッシュとがある。
・ディスクキャッシュを有効にするには、SMARTDRV.EXE を常駐させる。
・書き込みキャッシュをオフにするには/X スイッチを指定する。
・特定のドライブに対して書き込みキャッシュをオフにするにはドライブ名を指定する。
・特定のドライブに対してキャッシュ機能を無効にするにはドライブ名に「－」を付けて指定する。
・一部の SCSI ハードディスクでは、CONFIG.SYS でダブルバッファの組み込みが必要になる。
・書き込みキャッシュがオンの場合には、電源を切ったりリセットをする前には、5 秒程度待つか、/C スイッチを付けて実行する。

●解 説●

ディスクキャッシュのしくみ

CPUの処理速度は年々目まぐるしく向上しています。同じく、ディスク装置のアクセス速度も向上してはいますが、そのテンポはCPUほど速くはありません。そのため、CPUがディスクへの読み書き命令を出してから、それが実行されるまでの待ち時間は、どんどん長くなってきているのです。

こうした速度差の問題を緩和するため、DOS6には**SMARTDRV.EXE**というディスクキャッシュプログラムが付属になっています。ディスクキャッシュプログラムは、ディスクから読み出したデータをあらかじめ確保しておいたメモリへも記憶しておいて、次に同じ部分を読み出す指令があった際には、ディスクからではなく、メモリの内容を利用するというしくみになっています。CONFIG.SYSのコマンドにあったディスクバッファ（BUFFERS）と似ていますが、ディスクバッファはディスクの管理領域を記憶するのに対して、ディスクキャッシュは管理領域はもちろん、データまで記憶する点が異なります。つまり、もっと大掛かりなものなのです。

現在のシステムでは、ディスクキャッシュは必須と言えそうです。試しに、ディスクキャッシュを組み込まずにWindowsを起動してみると、その違いに驚くはずです。

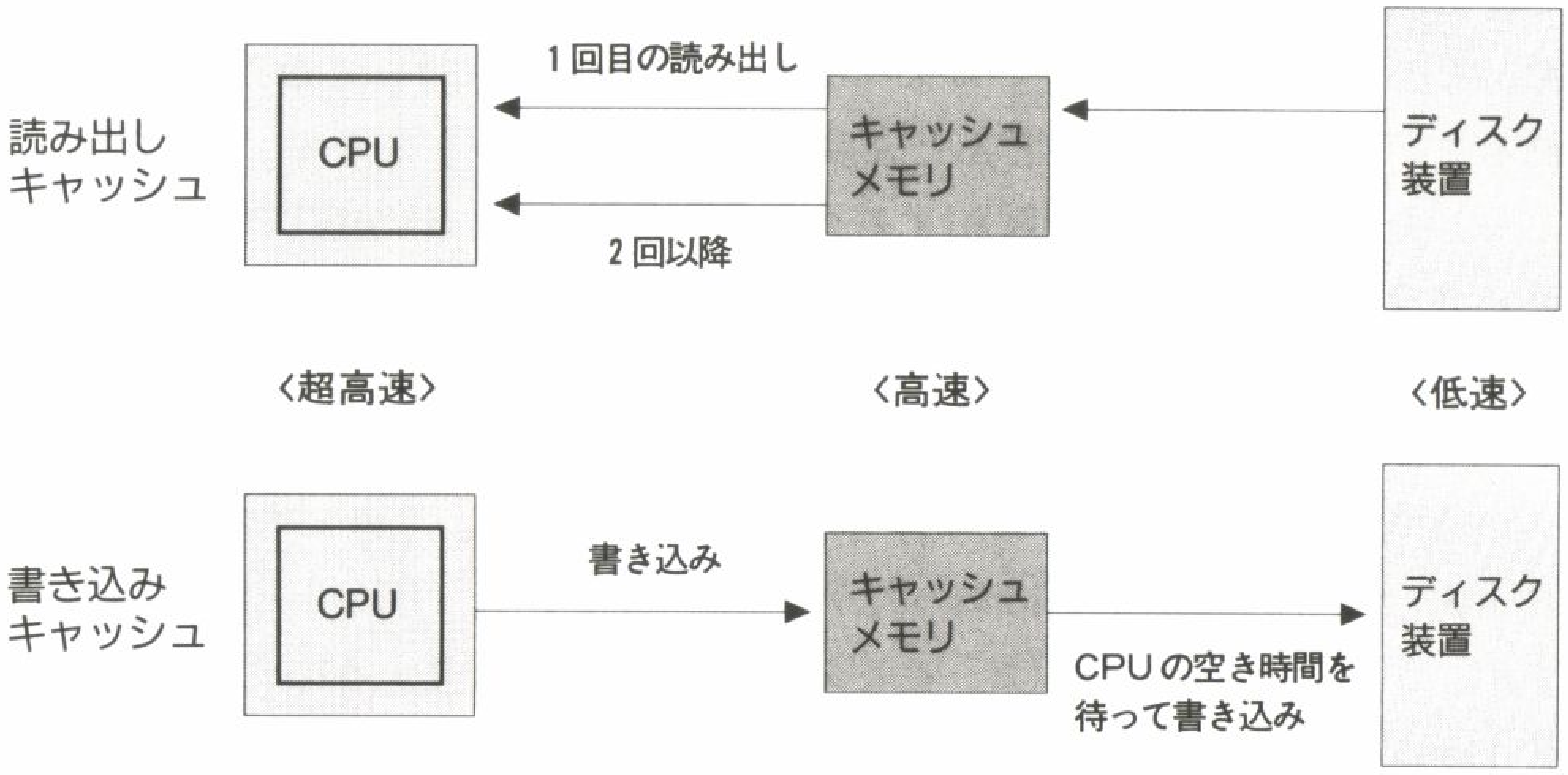
読み込みキャッシュと書き込みキャッシュ

DOS6に付属のSMARTDRV.EXEや、Windows3.1に付属のSMARTDRV.EXEには、ディスクから読み込んだ際にはたらく**読み込みキャッシュ（リードキャッシュ）**のほかに、ディスクへ書き込む際にはたらく**書き込みキャッシュ（ライトキャッシュ）**という機能もあります。

書き込みキャッシュは、ディスクへ書き込む指令があっても、直ちに処理するのではなく、CPUに空き時間ができるまでキャッシュメモリに蓄えておくというしくみです。ただし、いつまでも待っているわけではなく、たとえCPUが暇にならなくても、一定時間が経過すると書き込みは実行されます。

書き込みキャッシュは、絶大な効果がある反面、キャッシュ内に書き込みデータが残っているうちに電源を切ったり、システムをリセットしてしまうと、ファイルの内容が失われることになってしまいます。使用には、十分な注意が必要です。

●読み込みキャッシュと書き込みキャッシュのしくみ



SMARTDRV.EXEの組み込み方

SMARTDRV.EXEは、AUTOEXEC.BATで組み込みます。SMARTDRV.EXEの場合、十分なUMBメモリがあれば、自動的にUMBメモリに常駐するようになっています。そのため、LHコマンドを用いる必要はありません。

● SMARTDRV.EXEの組み込み

```
A:¥DOS¥SMARTDRV.EXE　初期キャッシュサイズ　Windowsキャッシュサイズ
```

初期キャッシュサイズというのは、SMARTDRV.EXEを実行した時点のキャッシュサイズのことです。Windowsを起動した場合、できる限り多くの拡張メモリを確保するため、**Windowsキャッシュサイズ**まで縮小されるようになっています。

初期キャッシュサイズとWindowsキャッシュサイズを指定しない場合は、拡張メモリの容量によって次のように設定されます。

◉キャッシュサイズ省略時の既定値

拡張メモリ	キャッシュサイズ	Windows キャッシュサイズ
1M バイトまで	すべての拡張メモリ	0（キャッシュはオフ）
2M バイトまで	1M バイト	256K バイト
4M バイトまで	1M バイト	512K バイト
6M バイトまで	2M バイト	1M バイト
6M バイト以上	2M バイト	2M バイト

キャッシュメモリが大きくなれば、それだけ多くのデータを記憶できて、ディスクをアクセスする回数も減ることになります。キャッシュメモリの最適な大きさは、使用しているハードディスクの容量と、搭載されている拡張メモリの大きさによって決めてください。最近は、大容量のハードディスクが多いため、数Mバイトのキャッシュメモリが望ましいと思います。ただし、Windows環境では拡張メモリの容量によって、起動できるアプリケーションの数や使い勝手が違ってくるため、6Mバイト程度の拡張メモリであれば、1Mバイト程度に抑えておくほうがよさそうです。

たとえば、初期キャッシュサイズを4Mバイトにして、Windowsキャッシュサイズを2Mバイトにするには、次のように指定します。

```
A:¥DOS¥SMARTDRV.EXE 4096 2048
```

なお、DOSのインストールプログラムを実行すると、SMARTDRV.EXEがAUTOEXEC.BATの先頭部分に記述されていることがあります。しかし、CD-ROMに対してディスクキャッシュを有効にするには、MSCDEX.EXEのあとにSMARTDRV.EXEを組み込む必要があるため、記述する位置を変更しておいてください。

ディスクキャッシュのオン／オフ

DOS6のSMARTDRV.EXEでは，ハードディスクに対しては最初から読み込みキャッシュと書き込みキャッシュの両方がオンで、フロッピーディスクドライブとCD-ROMドライブに対しては、読み込みキャッシュのみがオンになっています。

すべてのドライブに対して書き込みキャッシュのみをオフにしたい場合には、次のように/Xスイッチを付けて実行してください。

●全ドライブの書き込みキャッシュ「オフ」の設定例

```
A:¥DOS¥SMARTDRV.EXE 4096 2048 /X
```

また、特定のドライブに対して書き込みキャッシュのみをオフにしたい場合には、ドライブ名を指定します。

たとえば､A～Cドライブの書き込みキャッシュをオフにするには､次のように指定してください。

●指定ドライブの書き込みキャッシュ「オフ」の設定例

```
A:¥DOS¥SMARTDRV.EXE A B C 4096 2048
```

書き込みキャッシュがオフになっているドライブに対して、書き込みキャッシュをオンにするには、ドライブ名に「＋」を付けて指定します。

たとえば､F～Gドライブの書き込みキャッシュをオンにするには､次のように指定してください。

●指定ドライブの書き込みキャッシュ「オン」の設定例

```
A:¥DOS¥SMARTDRV.EXE F+ G+ 4096 2048
```

安全のため、フロッピーディスクに対しては書き込みキャッシュは用いないようにしましょう。

●特定ドライブのキャッシュ機能を無効にする

特定のドライブに対してキャッシュ機能を無効にするには、ドライブ名に「－」を付けて指定してください。

F、Gドライブのキャッシュをオフにするには、次のように指定します。

◉指定ドライブのキャッシュを無効にする設定例

```
A:¥DOS¥SMARTDRV.EXE F- G- 4096 2048
```

ダブルバッファ機能が必要な場合の設定

一部のSCSIインターフェイスを用いたハードディスクでは、SMARTDRV.EXEを使用する際に、**ダブルバッファ**という機能を組み込む必要が生じます。めったに必要とはならないのですが、SCSIインターフェイスでハードディスクを利用している場合には、最初に確認の作業を行ってください。

まず、CONFIG.SYSにダブルバッファの記述を加えておきます。

◉ダブルバッファの指定（CONFIG.SYS）

```
DEVICE = A:¥DOS¥SMARTDRV.EXE /DOUBLE_BUFFER
```

システムを再起動して、SMARTDRV.EXE をオプションなしで実行してみます。

◉ SMARTDRV.EXE の実行例

```
A:¥>SMARTDRV
Microsoft SMARTDrive ディスクキャッシュ Version 5.0
Copyright 1991,1993 Microsoft Corp.

キャッシュサイズ:  5,242,880 バイト
Windows 実行時のキャッシュサイズ:  3,145,728 バイト

          ディスクキャッシュの状態
ドライブ キャッシュ読み出し キャッシュ書き込み  バッファリング
---------------------------------------------
  A:        する          する          しない
  B:        する          する          しない
  C:        する          しない        しない
  D:        する          しない        しない
  F:        する          しない        しない
後書きデータはコマンドプロンプトに戻る前に書き込まれます.

"SMARTDRV /?" と入力するとヘルプが表示されます.
```

「バッファリング」の部分が「する」となっている場合には、そのままダブルバッファを組み込んで使用してください。「しない」となっていれば、CONFIG.SYS に加えたダブルバッファの記述を削除するか、あるいはコメントにしておきます。

いずれの場合も、AUTOEXEC.BAT における SMARTDRV.EXE の記述方法は、通常の場合と変わりません。

Part 2
実用編

実用編の内容

基礎編で学んだ知識を元にして、皆さんが実際に使っているシステムの環境に合ったCONFIG.SYSやAUTOEXEC.BATを作成していただければよいのですが、慣れないうちは戸惑うことも多いと思います。

そこで、実用編では、さまざまなシステム環境に応じたCONFIG.SYSやAUTOEXEC.BATの例をいくつか用意しました。まったく同じ環境ではなくても、なかには参考にしていただけるものもあるはずです。

なお、実用編で示しているCONFIG.SYSやAUTOEXEC.BATの内容は、すべてあとからMemMakerというユーティリティを用いて最適化の処理を行うことを前提としています。そのため、デバイスドライバをUMBメモリに組み込むDEVICEHIGHコマンドや、常駐プログラムをUMBメモリに常駐させるINSTALLHIGHコマンド、LOADHIGH(LH)コマンドは用いていません。

Step 32

CONFIG.SYS の記述順

●ポイント●

・基本コマンドはどのような順番で記述してもかまわない。
・デバイスドライバは、記述する順番を考慮する必要がある。

●解 説●

順番には関係ないコマンド

CONFIG.SYS に記述する内容には、順番を考慮する必要のあるものと、ないものとがあります。考慮しなくてもよいのは、次に挙げるコマンド類です。

BREAK	BUFFERS	DOS	FCBS	FILES
LASTDRIVE	REM(;)	SET	SHELL	

これらのコマンドは、実行されるタイミングが決まっているため、どこに記述してあってもかまわないのです。

デバイスドライバや常駐プログラムを組み込む順番

CONFIG.SYS でデバイスドライバや常駐プログラムを組み込む場合には、次の順番に注意してください。

◉ CONFIG.SYS コマンドの記述順

- まず HIMEM.SYS を、次に EMM386.EXE を記述する。
- HIMEM.SYS と EMM386.EXE は、拡張メモリを利用するほかのプログラム（RAM ディスク用ドライバなど）より先に記述する。
- KKCFUNC.SYS が必要な日本語 FEP を組み込むなら、FEP のドライバより前に KKCFUNC.SYS を記述する。

つまり、次の順番はほぼ決まっていると考えてよいでしょう。

◉順番が決まっているコマンドによる CONFIG.SYS の記述例

```
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=（日本語FEP）
```

決まっているのは順番だけであって、間にほかのドライバを記述するのはかまいません。ただし、HIMEM.SYS と EMM386.EXE だけは始めに記述しましょう。

Step 33

CONFIG.SYSとAUTOEXEC.BATの基本型

●ポイント●

- CONFIG.SYSやAUTOEXEC.BATの最初の設定はまちまちだが、最低限の記述はおさえておこう。
- BUFFERSの値は、ディスクバッファがHMAメモリに収まる範囲にしておくほうがよい。

●解　説●

最低限の記述をしたCONFIG.SYS

DOS6を新規にインストールした直後のCONFIG.SYSは、システム環境によって多少の違いがありますし、アップグレードセットを用いた場合には、基本的に以前の環境を引き継ぐようにCONFIG.SYSが作成されています。また、最近ではDOS6がハードディスクにインストール済みの機種も増えてきていますが、それらのCONFIG.SYSでも機種による違いがあります。

このように、CONFIG.SYSはスタートの時点ですでにバラバラになってしまっているため、本来なら、どれが基本型ということは言えないのです。しかし、「これだけ設定してあれば、まずほとんどのシステムで正常にDOS6が起動できる」という程度であれば、とりあえず考えられそうです。

それが、次に示したCONFIG.SYSです。

◉一般的なCONFIG.SYSの基本型

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
```

この記述が基本となって、さまざまな機能を付加するための記述が増えてゆくのだと考えてください。

● BUFFERSとFILESの値

通常、DOS6をインストールしたばかりのCONFIG.SYSでは、BUFFERSの値が20〜30程度になっていると思いますが、基礎編で解説したように、HMAメモリに収まる範囲を探して指定してください。3〜6程度になるはずです。

FILESの値は、通常20程度あれば十分です。しかし、データベースソフトや言語ソフトなど、特に多くのファイルを使用するソフトウェアを利用する場合には、それらのマニュアルを読んで、指定された値を設定しておいてください。インストールプログラムによって、自動的に変更される場合もあります。ここでは30にしていますが、これでたいていは足りるはずです。

● SHELLコマンドの設定

SHELLコマンドの設定は、COMMAND.COMをルートディレクトリに置いていない場合には絶対に必要です。その場合、再読み込み先まで、正しく記述するようにしてください。COMMAND.COMが起動ドライブのルートディレクトリにあるなら、SHELLコマンドを省いても特に問題はありません。しかし、環境領域の拡張をする場合などにも用いますから、とりあえず記述しておくほうがよいと思います。

● DOSコマンドと拡張メモリマネージャの設定

DOSコマンドは、決まり文句のように考えてもよいでしょう。

拡張メモリマネージャは、DOSに付属のHIMEM.SYS、EMM386.EXEを使う代わりに、市販のものを使うこともできます（それらについては応用編で解説)。なお、しばらく使っていて問題が起きなければ、起動時のメモリチェックを省くように、HIMEM.SYSに次のオプションを付けておくとよいでしょう。

```
DEVICE=A:¥DOS¥HIMEM.SYS /TESTMEM:OFF
```

最低限の記述をしたAUTOEXEC.BAT

AUTOEXEC.BATに関しても、CONFIG.SYSと同じことが言えるでしょう。つまり、システムによって最初の内容から異なってはいますが、最低限の設定というのは存在すると思います。

◉一般的なAUTOEXEC.BATの基本型

```
@ECHO OFF
PATH=A:¥;A:¥DOS
SET DOSDIR=A:¥DOS
SET TEMP=（作業用ドライブ名、ディレクトリ名）
PROMPT $P$G
```

● ECHOとPATHの設定

ECHOコマンドで「OFF」を指定しているため、以後のコマンド文字列は表示されなくなっています。ECHOコマンドの先頭に「@」を付けてあるので、ECHOコマンド自身も表示されません。

PATHコマンドで設定する検索パスは、インストールしているアプリケーションによって異なってきます。Windowsがインストールしてあれば、そのディレクトリも加えておいてください。最低でもバッチファイルのあるパスだけには検索パスの設定をしておかないと、たいへん面倒なことになってしまいます。

●環境変数の設定

環境変数 DOSDIR は、主に DOS シェルが参照する環境変数です。DOS シェルを使わないのであれば削除してしまってもかまいませんが、ほかのプログラムが参照しているかもしれないため、とりあえず設定しておくほうがよいと思います。

環境変数 TEMP（または TMP）は、RAM ディスクを作成してあれば、そのドライブを指定するようにしましょう。RAM ディスクがなければ、最も空き容量の大きいハードディスクのドライブなどを指定しておきます。

PROMPT コマンドでは「PG」を指定して、カレントディレクトリがわかるようにしておくほうが便利です。

AUTOEXEC.BAT でウィルスチェックを行う

AUTOEXEC.BAT 内で、DOS6 に付属の「ウィルスバスター 98」を使ったウィルスチェックを行っておくとよいでしょう。それには、AUTOEXEC.BAT の最後に、次の 1 行を加えておきます。

◉ウィルスチェックの設定

```
VB98 -V
```

これで、メモリと全ドライブに対してウィルスチェックが行われます。

決まった日時にウィルスチェックを行いたい場合には、まず VB98 のメニュー画面を起動して、メニューにある「予約検索設定」で設定を済ませておきます。そして、AUTOEXEC.BAT の最後に次のように記述してください。

◉決まった日時にウィルスチェックを行う

```
VB98 -SCHEDULE
```

DOS6 に付属のウィルスバスター 98 は、機能縮小版のため、ウィルスの検索機能だけしかありません。もしも、ウィルスが発見されたと表示された場合には、市販されている「ウィルスバスター 98」などのウィルス除去プログラムを購入するか、パソコン通信のセンターにあるライブラリからワクチンプログラムをダウンロードして、すみやかにウィルスを退治してください。

Step 34

市販の日本語 FEP を使用する

●ポイント●

・ATOK8 では、KKCFUNC.SYS は必要ではない。
・ATOK8 では、DPMI に対応したドライバを組み込んでいるなら、ATOK8A.SYS に/Z=1 の指定が必要。
・ATOK8 では、DOS シェルのタスクスワップ機能を利用するなら、ATOK8EX.SYS が必要。
・WX3 では、KKCFUNC.SYS が必須。
・VJE-Delta は、DOS 環境、Windows 環境、Windows の DOS ボックスで動作する。
・VJE-Delta は VCPI 対応の拡張メモリマネージャを組み込み、DPMI サーバを起動させた状態で組み込む必要がある。
・VJE-Delta では、KKCFUNC.SYS は必要ではない。

●解 説●

ATOK8を組み込む

DOS環境で利用する代表的な日本語FEPのひとつ『**ATOK8**』は、ワープロソフトのベストセラー『一太郎 Ver.5』に付属になっているほか、単体でも販売されています。

ATOK8を組み込んだ場合のCONFIG.SYSは、次のようになります。

◉ ATOK8を組み込んだCONFIG.SYSの例

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8B.SYS
```

ATOK8では、**ATOK8A.SYS**、**ATOK8B.SYS**という2つのドライバが必要です。特にオプションを指定しなくても、EMSメモリを利用して常駐サイズを小さくするようになっています。そのため、必ず拡張メモリマネージャよりあとに記述してください。

日本語変換補助ドライバのKKCFUNC.SYSは、複数のFEPを組み込んで、切り替えながら使うということをしなければ、組み込まなくてもかまいません。

ATOK8A.SYSに付いている/UCFスイッチは、インストール時に自動的に付加されます。このスイッチにより、**ATUT.EXE**という環境設定ユーティリティの設定内容が保存されている環境ファイルの指定を行います。

ATOK8をDPMIに対応した拡張メモリマネージャといっしょに利用するなら、ATOK8A.SYSに/Z=1というオプションを付けてください(DPMIの設定については後述します)。

● DOS シェルのタスクスワップを使う場合の設定

ATOK8A.SYS、ATOK8B.SYS のほかに、DOS シェルのタスクスワップ機能を利用するなら、**ATOK8EX.SYS** というドライバを組み込んでください。

◉タスクスワップを使う場合の ATOK8 の組み込み

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB /T=A:¥DOS¥EXTDSWAP.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
```

● ADDDRV コマンドで ATOK8 を組み込む場合

もし ADDDRV コマンドを使うなら、CONFIG.SYS から ATOK8 のドライバ類の記述を削除しておきます。

◉ CONFIG.SYS から ATOK8 の記述を削除する

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
```

そして、次のような内容で ATOK8.DEV というファイルを作成しておきます。

◉ ADDDRV 用 ATOK8 組み込みファイルの例

```
DEVICEHIGH=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICEHIGH=A:¥ATOK8¥ATOK8B.SYS
```

仮にATOK8.DEVをATOK8というディレクトリに作成したなら、AUTOEXEC.BATの最後で次のように記述するか、またはDOSの起動後に実行してください。

◉ ADDDRVによるATOK8組み込みの実行

```
ADDDRV A:¥ATOK8¥ATOK8.DEV
```

これで、ATOK8が使用できるようになります。不要になったら、DELDRVコマンドで常駐を解除させてください。

ただし、FEPのように比較的常駐サイズの大きなドライバは、CONFIG.SYSで組み込んでおくほうが、メモリを効率よく使うことができます。

WX IIIを組み込む

『WX III』は、カラーカスタマイズ、キーカスタマイズ、ローマ字カスタマイズなどの機能をいちはやく取り入れた日本語FEPです。主なFEPのキー操作や表示スタイルをまねる機能もあるので、ほかのFEPからの移行はたいへん容易だと思います。

●カーネルとフロントエンド部の組み込み

WX IIIでは、カーネル部と呼ばれる**WXK.SYS**と、フロントエンド部と呼ばれるドライバを組み合わせて使用します。フロントエンド部は、どのFEPに似せたシステムにするかという選択によって異なります。

◉ WX IIIのフロントエンド部

- WX3.SYS　　　：WX IIIオリジナルシステム
- WX3A6.SYS　　：ATOK6ライクなシステム
- WX3A7.SYS　　：ATOK7ライクなシステム
- WX3A8.SYS　　：ATOK8ライクなシステム
- WX3VJ(2).SYS：VJE-β(2)ライクなシステム
- WX3MT.SYS　　：松茸V3ライクなシステム

また、内部から WX3D.SYS、WXKC.SYS というファイルを呼び出しているので、それらのファイルも同一のディレクトリに入れておく必要があります。

WX IIIは、必ず KKCFUNC.SYS と一緒に組み込むようにしてください。たとえば、オリジナルのシステムで使用するなら、次のような CONFIG.SYS になります。

◉ WX IIIと KKCFUNC.SYS を組み込む

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥WX3¥WXK.SYS /A5
DEVICE=A:¥WX3¥WX3.SYS /INI=A:¥WX3¥WX3SYS.INI /A5
```

WXK.SYS と **WX3.SYS** の両方に付いている/A5 というスイッチは、EMS メモリと UMB メモリに余裕があれば、自動的に使用せよという指定です。したがって、WXK.SYS やフロントエンド部（WX3.SYS など）は、DEVICEHIGH コマンドで組み込まなくても UMB メモリに読み込むことができるのです。DEVICEHIGH コマンドを用いると、UMB メモリが足りないというエラーで WX3 が組み込まれない場合もあるので注意してください。

フロントエンド部の/INI スイッチでは、設定ファイルを指定します。

●機能拡張オプションドライバの組み込み

WX IIIには、機能拡張のためのオプションドライバも用意されています。

◉ WX IIIのオプションドライバ

- WX3TX.SYS ：オプション単漢字変換システム
- WX3TM.SYS ：オプション単漢字入力システム
- WX3TGG.SYS：オプション単語登録（ガイド入力）システム

これらのドライバは、好みによって組み込んでください。一部だけでもかまいません。

◉オプションドライバの組み込み例

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥WX3¥WXK.SYS /A5
DEVICE=A:¥WX3¥WX3.SYS /INI=A:¥WX3¥WX3SYS.INI /A5
DEVICE=A:¥WX3¥WX3TX.SYS /A1
DEVICE=A:¥WX3¥WX3TM.SYS /A1
DEVICE=A:¥WX3¥WX3TGG.SYS
```

WX3TX.SYS と **WX3TM.SYS** にある/A1スイッチは、EMSメモリを利用するという指定です。

● ADDDRV コマンドで WX III を組み込む場合

ADDDRVコマンドを使うなら、CONFIG.SYSからKKCFUNC.SYSとWX IIIのドライバ類の記述を削除しておきます。

◉ CONFIG.SYS から WX IIIの記述を削除する

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
```

そして、次のような内容でWX3.DEVというファイルを作成しておきます。

◉ ADDDRV用WX III組み込みファイルの例

```
DEVICEHIGH=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥WX3¥WXK.SYS /A5
DEVICE=A:¥WX3¥WX3.SYS /INI=A:¥WX3¥WX3SYS.INI /A5
```

仮にWX3.DEVをWX3というディレクトリに作成したなら、AUTOEXEC.BATの最後で次のように記述するか、またはDOSの起動後に実行してください。

◉ ADDDRVによるWX III組み込みの実行

```
ADDDRV A:¥WX3¥WX3.DEV
```

これで、WX IIIが使用できるようになります。不要になったら、DELDRVコマンドで常駐を解除させてください。

ただし、FEPのように比較的常駐サイズの大きなドライバは、CONFIG.SYSで組み込んでおくほうが、メモリを効率よく使うことができます。

VJE-Deltaを組み込む

従来の日本語FEPは、DOS環境に対応したものと、Windowsに対応したものとが明確に分けられていました。しかし、この『**VJE-Delta**』(Ver.1.1)は、DOS環境、Windows環境、それからWindowsの**DOSボックス**(Windows内から起動するDOS環境のこと）の3つの環境すべてに対応しています。

VJE-DeltaをDOS環境またはWindowsのDOSボックス内で利用する場合、VCPIをサポートしている拡張メモリマネージャを組み込んだ上で、DPMIサーバが起動された状態で組み込むという条件があります。そのため、通常のFEPを組み込むような設定では利用できません。拡張メモリマネージャとしてVMM386.EXEが付属になっていますが、ここでは、DOS6に付属の拡張メモリマネージャのもとで利用する方法について解説します。

なお、VMM386.EXEは、「Part4 Other Tools編」で解説する『MEMORY Server II』に付属の拡張メモリマネージャとほぼ同じ使い方になりますから、そちらの解説も参考にしてください。

●/DPMIスイッチとDPMI32.EXEの設定がポイント

VJE-DeltaをWindows環境からインストールすると、インストール先のディレクトリ内にCONFIG.VJE、AUTOEXEC.VJEというファイルが作成されているはずです。それらのファイルを参考にして、CONFIG.SYSとAUTOEXEC.BATを作成してください。基本型を元にすると、次のCONFIG.SYSになります。

◉ VJE-Deltaを組み込むCONFIG.SYSの例

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB /DPMI
DEVICE=A:¥VJEDELTA¥DPMI32.EXE
DEVICE=A:¥VJEDELTA¥VJEDDI.EXE /P=A:¥WINDOWS
```

ここでのポイントは、EMM386.EXEに/DPMIスイッチを付けておくことです。/DPMIスイッチは、本来DPMIを組み込み可能にするための指定ですが、同時にVCPIへの対応も行われます。

DPMI32.EXEは、VJE-Deltaに付属になっています。DOS6に付属のDPMI.EXEをDEVICEコマンドを用いて組み込んでも動作しますが、**VJEDDI.EXE**の常駐サイズが大きくなってしまうため、必ずDPMI32.EXEのほうを利用してください。

KKCFUNC.SYSを組み込むなら、必ずVJEDDI.EXEより先に記述しておきます。ここでは省いてありますが、VJE-Deltaを使っていて何らかの不具合が生じた場合には、KKCFUNC.SYSを組み込むと状況が改善されることがあるので、試してみてください。

VJEDDI.EXEに付いている/Pスイッチでは、Windowsがインストールされているディレクトリを指定します。

● ADDDRV コマンドで VJE-Delta を組み込む場合

ADDDRV コマンドを用いるなら、CONFIG.SYS から DPMI32.EXE、(KKCFUNC.SYS、) VJEDDI.EXE の記述を削除しておきます。

◉ CONFIG.SYS から VJE-Delta の記述を削除する

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB /DPMI
```

そして、次の内容で VJE.DEV というファイルを作成しておきます (KKCFUNC.SYS を組み込むことも可能です)。

◉ ADDDRV 用 VJE-Delta 組み込みファイルの例

```
DEVICEHIGH=A:¥VJEDELTA¥DPMI32.EXE
DEVICEHIGH=A:¥VJEDELTA¥VJEDDI.EXE /P=A:¥WINDOWS
```

VJE.DEV を VJEDELTA というディレクトリに作成したなら、AUTOEXEC.BAT の最後で次のように記述するか、または DOS の起動後に実行してください。

◉ ADDDRV による VJE-Delta 組み込みの実行

```
ADDDRV A:¥VJEDELTA¥VJE.DEV
```

これで、VJE-Delta が使用できるようになります。不要になったら、DELDRV コマンドで常駐を解除させてください。

VJEDDI.EXE の常駐サイズは 4K バイト程度と小さいのですが、DPMI32.EXE と一緒に組み込む必要があるため、CONFIG.SYS 内で組み込んでおくほうがメモリを効率よく使うことができると思います。

Step 35

RAM ディスクとディスクキャッシュを使用する

●ポイント●

- RAMDISK.SYS には割り当てる容量（128～14848 の範囲から 128 の倍数）を指定する。
- RAM ディスクのドライブは環境変数 TEMP または TMP に設定しておくとよい。
- SMARTDRV.EXE には割り当てる容量を指定する。
- 書き込みキャッシュをオンにしているのなら、電源を切る前に書き込み処理が行われたかどうかの確認が必要。
- 16M バイトを超えるメモリを搭載する場合には、システム空間の設定に注意する。

●解 説●

RAM ディスクを組み込む

RAM ディスクは、メモリの一部をディスク装置のように利用できるようにする機能です。たいへん高速であるというほかに、システムの電源を切ったりリセットすると、内容が完全に消去されるという特徴があります。そのため、プログラムが一時的に作業用ファイルを作成するドライブに指定するのには最適なのです。

また、ノート型パソコンをバッテリーで使う場合には、RAM ディスク内にプログラムとデータ、それから FEP の辞書ファイルをコピーしておくと、ハードディスクにアクセスしない分だけ、長い時間使用できるようになります。

最近のシステムでは、Windows 環境のために数 M バイト～数十 M バイトというメモリを搭載することが多くなっています。しかし、DOS 環境ではそれほど多くの拡張メモリは必要ないため、余った分を RAM ディスクに割り当てるとよいと思います。

RAM ディスクドライバの **RAMDISK.SYS** は、CONFIG.SYS で次のように記述してください。

◉ RAMDISK.SYS の組み込み例

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥DOS¥RAMDISK.SYS 6144
```

DOS 環境で RAM ディスクを利用するのなら、RAMDISK.SYS で必要なオプションは、容量（K バイト）の指定ぐらいなものと考えてよいでしょう。これは、128～14848 の範囲で、128 の倍数で指定してください。この例では、6144K バイト（6M バイト）を割り当てています。

●/E スイッチによるメモリ割り当て元の変更

このように設定すると、EMB メモリから割り当てられますが、/E スイッチを付けて EMS メモリから割り当てることもできます。どちらかというと、EMB メモリから割り当てるほうが高速にアクセスできるようになります。

ただし、EMS メモリは 16K バイトのページ単位で割り当てることができるため、割り当て容量の微調整が可能です。また、EMB メモリから割り当てる場合よりも比較的安全に確保できるため、EMB メモリを指定してエラーになったり、ストップしてしまった場合などには、/E スイッチを付けるとうまくゆくことがあります。

● RAM ディスクを活用するための設定

RAM ディスクを作成したら、SET コマンドを用いて環境変数 **TEMP** または **TMP** の設定も行っておきましょう。たとえば、F ドライブが RAM ディスクになるのなら、次のように設定します。

```
SET TEMP=F:¥
```

特に、日本語 FEP で使われる辞書ファイルを RAM ディスクにコピーしておくと、変換中にディスクをアクセスしないのでたいへん快適になります。AUTOEXEC.BAT のなかで、次のように記述しておくとよいでしょう。

```
COPY /B A:¥ATOK8¥*.DIC F:¥
```

「ATOK8」の部分は、使用している FEP に合わせて変えてください。/B スイッチは、その後のファイルをすべてバイナリファイルとして扱うための指定です。これがないと、途中にファイルの終了記号が混ざっているファイルでは、そこまでしかコピーが行われなくなってしまうのです。

ここでは、拡張子が「～.DIC」のすべてのファイルをコピーしていますが、一部の辞書ファイルだけを利用するなら、それらを並べて指定します。

このようにして使う場合、FEP の環境設定ユーティリティを用いて、辞書ファイルが RAM ディスクにあるという指定をしておくことも忘れないでください。

なお、RAM ディスクの内容が消えるとともに単語登録や辞書学習の結果も消えてしまいますから、最後に辞書の書き戻しを行うことも必要です。次のようなバッチファイルを作成しておくとよいでしょう。

```
COPY /B F:¥*.DIC A:¥ATOK8
```

ディスクキャッシュを組み込む

今や、なくてはならない存在になっているのが、**ディスクキャッシュ**です。ディスクキャッシュを組み込まないシステムでは、アプリケーションの起動時間などが大幅に違ってきますから、RAM ディスクなどとのバランスを考えながら、なるべく多くのメモリを割り当てるようにしましょう（とは言っても、10M バイト程度を限度にしてください）。

ディスクキャッシュプログラムの**SMARTDRV.EXE**は、AUTOEXEC.BATで次のように記述してください。

◉ SMARTDRV.EXEの組み込み例

```
@ECHO OFF
PATH=A:¥;A:¥DOS
SMARTDRV 3072
SET DOSDIR=A:¥DOS
SET TEMP=(作業用ドライブ名、ディレクトリ名)
PROMPT $P$G
```

SMARTDRV.EXEも、RAMディスクと同様に割り当てる容量を指定するだけで十分です。ここでは3072Kバイト（3Mバイト）を割り当てています。UMBメモリに余裕があれば、自動的に利用するようになっているため、LOADHIGH(LH)コマンドで組み込む必要はありません。

Windowsを起動することがあれば、次のようにWindows起動時のサイズを指定しておくとよいでしょう（詳しくは「Step41 Windowsを起動するための設定」で解説します）。

```
SMARTDRV 3072 2048
```

SMARTDRV.EXEを利用する場合、基礎編で解説したダブルバッファの確認を忘れないようにしてください。

書き込みキャッシュ利用時の注意

SMARTDRV.EXEで書き込みキャッシュを有効にしている場合、キャッシュメモリ内にデータが残っているうちに電源を切ってしまったり、システムをリセットしてしまうと、大切なデータが失われてしまいます。

書き込みキャッシュは、5秒以上経過すると自動的に書き込みが行われるようになっています。電源を切ったり、システムをリセットする前には、5秒以上待つようにしましょう。

また、次のように/CスイッチをつけてSMARTDRV.EXEを実行すると、強制的に書き込みが行われます。

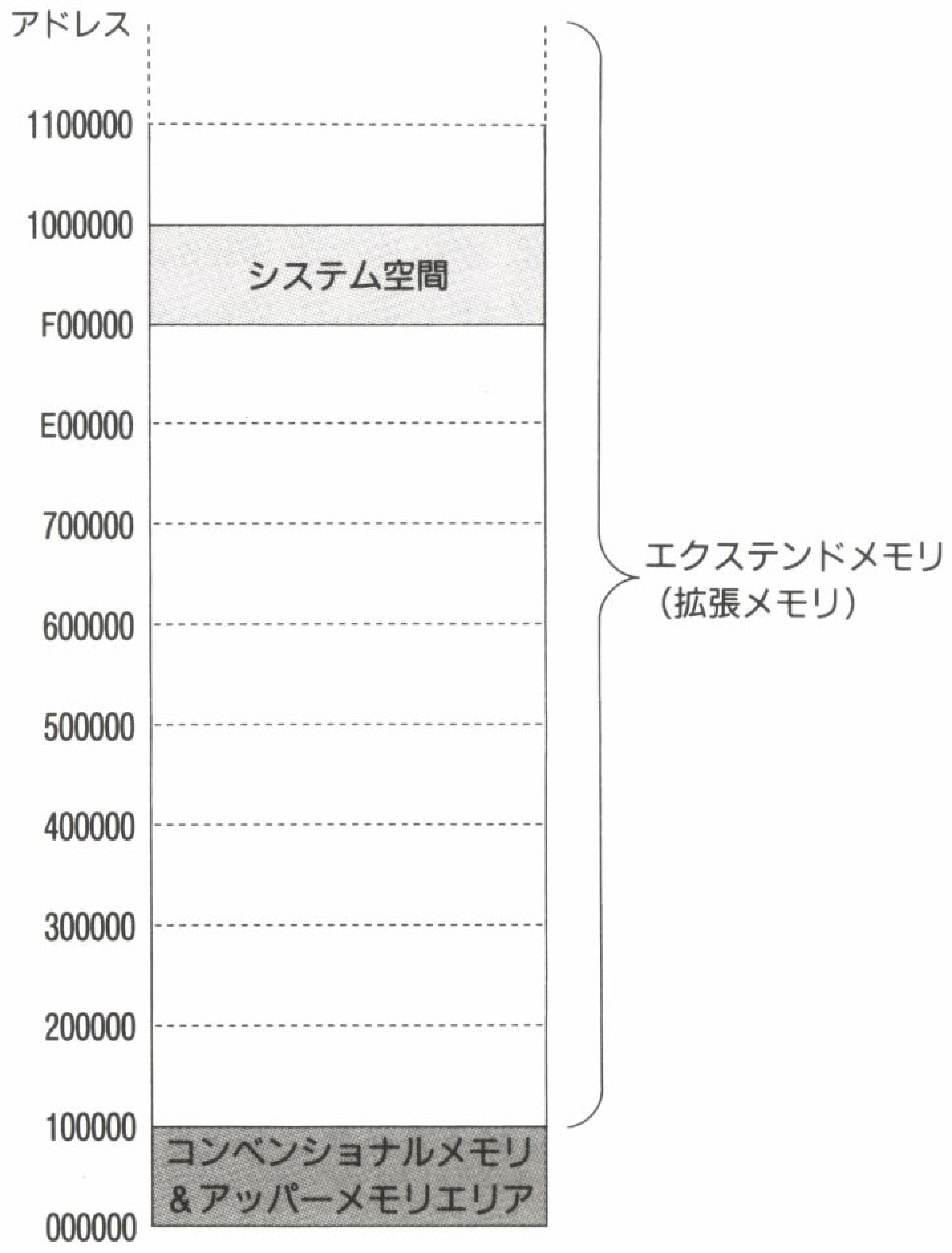
```
A:¥>SMARTDRV /C
```

拡張メモリ使用時の注意

メモリを16Mバイト以上搭載しているシステムで、RAMディスクやディスクキャッシュを組み込む場合には、**システム空間**（アドレスF00000h～FFFFFFh）に注意してください。

PC-9800シリーズでは、もともと搭載できる最大のメモリは14.6Mバイトが上限になっていたため、メモリの15Mバイト～16Mバイトにあたる部分はシステム空間として、システムが利用可能ということにしていたからです。

◉ PC-9800シリーズのメモリの様子



最近の機種では、このメモリの制限はなくなりましたが、Windowsで利用するグラフィックアクセラレータなどオプションボードの一部は、相変わらずシステム空間を利用しているため、あらかじめこの領域をシステムのために確保しておかなければなりません。

●RAMディスクなどがシステム空間を使わないようにする

14.6Mバイトを超えるメモリを搭載可能な機種では、システム設定のメニューに「**16Mバイトシステム空間**」という項目があります。ただし、ここを【使用する】に設定すると、その部分で拡張メモリが分断され、しかも、拡張メモリの容量が1Mバイト減ってしまうことになります。

【切り離す】に設定すると、搭載されているすべての拡張メモリを利用できるようになりますが、グラフィックアクセラレータが当たり前のように使われている状況ですから、ほとんどのケースでは【使用する】に設定するしかないはずです。

【使用する】の状態では、RAMディスクやディスクキャッシュの設定で、システム空間によって分断された拡張メモリのブロックの大きさを超えた容量を指定すると、正常に組み込めないので注意してください。

この範囲を超えてRAMディスクを確保しようとすると、

```
RAM DISK 用のメモリが確保できません
RAM DISK は使用できません
```

というエラーメッセージが表示されます。この場合、RAMディスクが使えないだけではなく、メモリだけはしっかり確保された状態になっていることもあり、そうなるとメモリがまったくの無駄になってしまいます。

ただ、システム空間の位置から考えると、RAMディスクやディスクキャッシュに13〜14Mバイトもの容量を割り当てようとしないかぎり、こうした問題は起こらないはずです。

なお、DEVICEHIGHコマンドを用いてRAMDISK.SYSをUMBメモリに読み込もうとした際、UMBメモリが不足していると同様のエラーになることがあります。その場合には、DEVICEコマンドを用いて組み込むようにしてください。

Step 36

SCSIインターフェイスで CD-ROM装置/MO装置/PD装置を使用する

●ポイント●

・SCSI機器を使用する際には、SCSI IDの設定に注意が必要。
・CD-ROM装置を使用するためには、CONFIG.SYSでデバイスドライバを組み込み、さらにAUTOEXC.BATでMSCDEX.EXEを実行する。
・CD-ROM装置のデバイスドライバで指定したデバイス名とMSCDEX.EXEのデバイス名を一致させる必要がある。
・CD-ROM用のドライブを確保するために、CONFIG.SYSでLASTDRIVEコマンドを実行しておく。
・MS-DOS Ver.6.2に付属のプログラム(FORMAT.EXEなど)は、3.5インチMOでは128Mバイトタイプのみに対応している。
・MO装置を使用するには、ハードウェア的にドライブとして認識させる方法と、デバイスドライバを組み込む方法とがある。
・ASPIマネージャを利用すれば、原則的に異なるメーカーの機器でも使用できる。
・PD装置は、MO+CD-ROMという感覚で設定すればよい。
・PC-9801-100やPC-9821X-B02では、PDはデバイスドライバで使うことはできない。

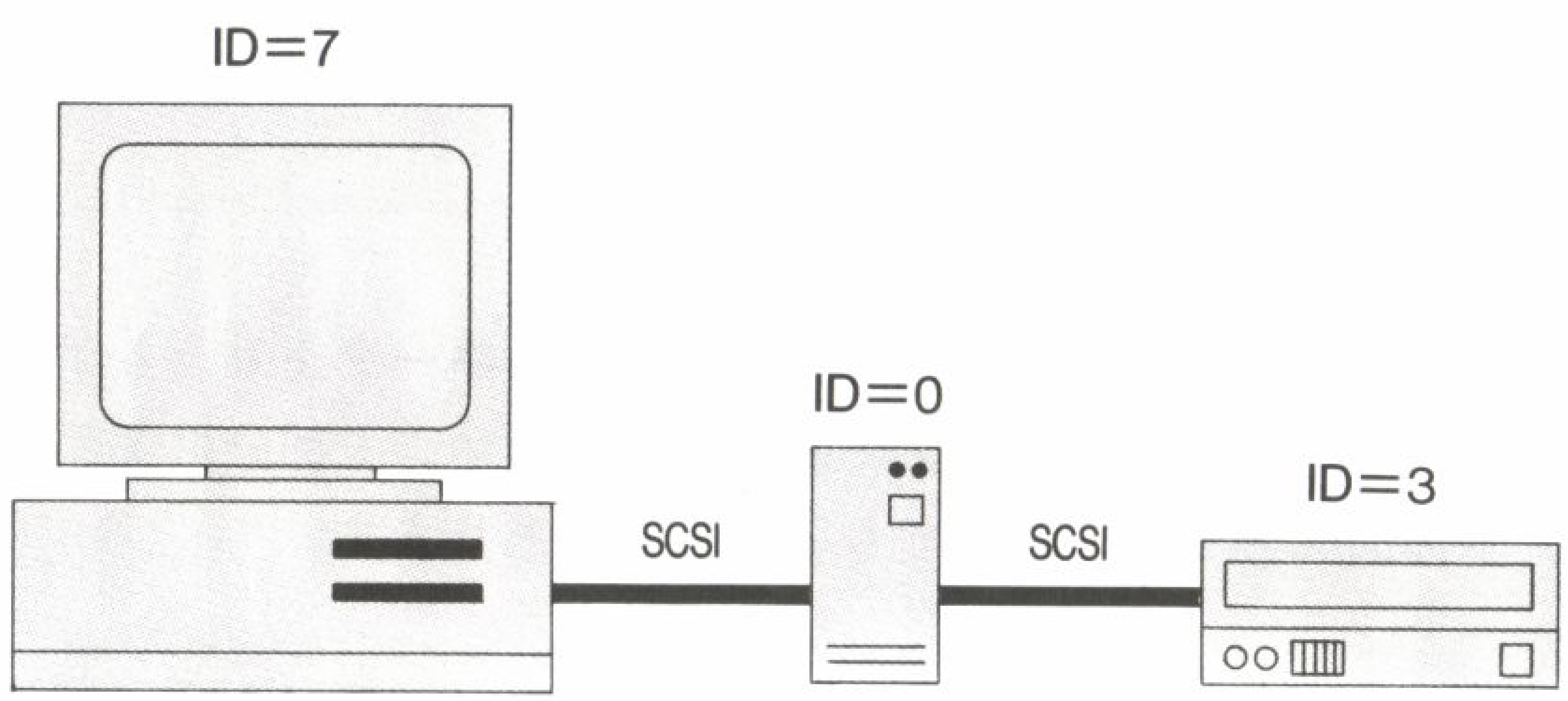
※「SCSI」は“スカズィー”と読まれることが多いようです。

●解 説●

SCSI 機器の接続は SCSI ID の設定に注意

SCSI インターフェイスに複数の装置を接続する場合、各装置のスイッチで **SCSI ID** と呼ばれる 0～7 までの番号を設定することになりますが、その番号の選択には特に注意が必要です。

◉ SCSI ID の設定例



通常、7 番にはパソコン本体（SCSI インターフェイスボード）を割り当て、0 番にはシステムを起動するハードディスクまたは MO 装置を割り当てるということはほぼ決まっています。問題は、その他の番号です。

筆者が使っている MO 装置では、デバイスドライバを用いる場合には、SCSI インターフェイスで接続しているハードディスクの SCSI ID と連続させないように指定するようになっています。たとえば、ハードディスクに 0 番を割り当てているなら、MO 装置には 1 番を割り当ててはいけないのです。逆に、デバイスドライバなしで用いるには、ハードディスクの SCSI ID と連続させなければなりません。

SCSI IDと周辺機器との関係をまとめると、次のようになります。

◉ SCSI IDと周辺機器の関係

- ハードディスク装置は、SCSI IDの0〜6までのどこに指定してあっても認識される（連続していなくてもよい）。
- MOやPD装置をデバイスドライバなしで利用する場合、SCSI IDが連続した状態になっている必要がある。
- MOやPD装置をデバイスドライバを組み込んで利用する場合、SCSI IDが連続していなくてもよい（連続させてはいけないと指定されている場合もある）。
- CD-ROM装置は、SCSI IDを連続させる必要はないが、0番は指定しないようにする。

それから、接続している最後の装置には、**ターミネータ**をセットすることも忘れないでください。ターミネータに関しては、コネクタ部分に部品を接続する製品と、本体のスイッチで設定できる製品とがあります。

CD-ROMドライバの組み込み

マルチメディアパソコンと呼ばれる機種では、**CD-ROM装置**が標準装備されていますが、その他の機種にとっても、今やCD-ROM装置はなくてはならない周辺機器になっていると思います。

CD-ROM装置は、ハードディスクのように、単にインターフェイスボードに接続しただけでは認識されないようになっています。まず、CONFIG.SYSでデバイスドライバを組み込み、次に、AUTOEXEC.BAT内で**MSCDEX.EXE**（**Microsoft CD-ROM Extensions**）を実行しなければなりません。

CONFIG.SYSで組み込むドライバは、CD-ROM装置に添付されているものを使ってください。ただし、DOS6にはNEC製のCD-ROM装置のためのドライバが付属していて、DOSの組み込み時に接続が確認されると、自動的にインストールされるようになっています。

●デバイス名の指定

どのメーカーのドライバでも、指定すべきオプションはほとんど同じと言ってよいでしょう。たとえば、DOS6に付属の**NECCD.SYS**を用いた場合のCONFIG.SYSは次のようになります。

◉ NECCD.SYSを使った場合のCONFIG.SYS

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
LASTDRIVE=Z
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥DOS¥RAMDISK.SYS 6144
DEVICE=A:¥DOS¥NECCD.SYS /D:CD_101
```

CD-ROM装置用のドライバでは、**デバイス名**の指定が重要になります。NECCD.SYSでは、/Dスイッチによって指定しますが、このほかのドライバも、ほぼ同じ仕様になっているはずです。ドライバ名はハードディスクやフロッピーディスクのラベル名だと思って好きな文字列を指定してください。マニュアルで用いられている用例と同じものにしておけば安心です。

● LASTDRIVEの指定

CD-ROM装置に付属のインストールプログラムを実行すると、たいていはCONFIG.SYSに、

```
LASTDRIVE=Z
```

という設定が書き込まれるはずです。これは、どのようなシステムでもCD-ROM用のドライブを確保できるようにするための処置です。PC-9800シリーズでは、接続されているハードディスク、フロッピーディスク、MO(光ディスク)などをハードウェア的にドライブとして認識するようになっていますが、CD-ROM装置はソフトウェア(デバイスドライバ)を用いないとドライブとして認識されません。そのため、LASTDRIVEコマンドによって使用できるドライブ数をハードウェアが認識した数よりも増やしておく必要があるのです。

実際には、CD-ROM 装置を含めたドライブ数を確保しておけば十分ですから、必要なドライブ数までに変更しておくのもよいでしょう。

● MSCDEX.EXE の実行

AUTOEXEC.BAT で実行する MSCDEX.EXE については、すべてのメーカーで共通ですが、DOS6 に付属の MSCDEX.EXE とタイムスタンプ(更新日時) を比較して、新しいほうを使うようにするほうがよいと思います。

MSCDEX.EXE では、CONFIG.SYS のドライバで指定したデバイス名をそのまま指定することが重要なポイントとなります。

◉ MSCDEX.EXE の設定例

```
MSCDEX /D:CD_101 /E
```

● MSCDEX.EXE のオプションスイッチ

デバイス名は、/D スイッチで指定します。

/E スイッチは、EMS メモリを使用するという指定です。これがないと常駐サイズがたいへん大きくなってしまうため、CD-ROM 装置に付属のインストールプログラムによって付けられていなければ、自分で付けるようにしてください。

通常、CD-ROM 装置のドライブ名は、MSCDEX.EXE の実行時に使われている最終ドライブの次になりますが、特に指定したければ/L スイッチを用いてください。たとえば、F ドライブにしたければ、

```
MSCDEX /D:CD_101 /E /L:F
```

とします。ただし、この場合には CONFIG.SYS で LASTDRIVE コマンドを用いてドライブ数を拡張しておくことが必要となります。

また、CD-ROM 装置によっては**セクタバッファ**の値を変更しないと、正常に読み込むことができないといった不具合を生じることがあります。もしも、ドライブを認識しているにもかかわらず読み込みに失敗することがあれば、/M スイッチで 15〜20 程度の値を指定してみてください。

```
MSCDEX /D:CD_101 /E /L:F /M:15
```

CD-ROM 装置は、SCSI インターフェイス以外にも、IDE インターフェイスやサウンドボード付属のインターフェイスによって接続が可能ですが、いずれの場合でもドライバと MSCDEX.EXE を組み合わせて用います。

MO(光ディスク)を使用する2つの方法

MO（光ディスク）は、外見上 CD-ROM とよく似ていますが、CD-ROM が読み出し専用なのに対して、MO は書き込みが可能という違いがあります。5 インチサイズの MO は、CD-ROM と同じ 650M バイト程度の容量があります。しかし MO のフォーマットの規格などがメーカーによってまちまちなため、現在のところあとから開発された 3.5 インチサイズの MO 装置のほうが多く使われています。その 3.5 インチ MO 装置には、128M バイト用と 230M バイト用とがあり、230M バイト用の製品のほとんどは 128M バイトの MO も読み書き可能となっています。

MS-DOS Ver.6.2 では、FORMAT コマンドなどが 128M バイトタイプの MO のみに対応しているため、230M バイトタイプの MO をフォーマットするには、MO 装置に付属のフォーマットプログラムが必要になります。ただし、フォーマットされた 230M バイトタイプの MO の読み書きは正常に行われます。

●ドライバの有無で使い方が分かれる

MO 装置を使用するには、2 種類の方法があります。1 つは、デバイスドライバを用いる方法、もう 1 つは、単に接続するだけでハードウェア的にドライブとして認識させる方法です。

ただし、ハードウェア的に認識させるためには、

①使用している SCSI インターフェイスボードが PC-9801-55、PC-9801-92 などの MO 装置に対応したものであること
② MS-DOS Ver.5 以降を使用していること

が条件となります。MO 装置をどのようにして使ったらよいかということについては、SCSI インターフェイスと MO 装置のマニュアルをよく読んで判断してください。

デバイスドライバなしで使用すると、MO 装置からシステムが起動できるというメリットがあります。これは、複数の OS を使い分けたいような場合にはたいへん便利だと思います。

一方で、MO をセットした状態で電源を入れると、MO のほうにシステムを読み込みにいってしまうため、いちいち MO を出しておかなければならないという不便さが生じます。

デバイスドライバを用いると、MO をセットした状態で電源を入れても大丈夫ですが、当然のことながらドライバを組み込む分だけメモリが圧迫されてきます。ほかに多くのドライバ類を組み込んでいて、メモリに関して不安があるのであれば、デバイスドライバ抜きにするほうがよさそうです（ただし、可能であればですが)。

ASPI マネージャとデバイスドライバで MO 装置を使用する

最近、PC-9800 シリーズでも **ASPI マネージャ**が使われることが増えてきています。ASPI マネージャとは、Advanced SCSI Programming interface の略で、SCSI インターフェイスボードの違いを吸収してしまうプログラムのことです。従来、SCSI インターフェイスボードの仕様はメーカーによってまちまちで、装置のメーカーが異なると使用できないということがありました。そのような不便さを解消しようというのがねらいです。

DOS/V 機 (PC/AT 互換機) の世界では、以前から ASPI マネージャが使われていたため、どのメーカーの SCSI 装置を購入しても、まず間違いなく使用できるようになっているのです。

ASPI マネージャは、SCSI インターフェイスボードあるいは接続する装置に付属しています。ASPI マネージャは、SCSI インターフェイスボードによってプログラムが異なりますから、マニュアルをよく読んで、間違いなくインストールを行ってください。

たとえば、筆者が使用している Panasonic 製の MO 装置に付属の ASPI マネージャでは、PC-9801-55/92 互換の SCSI インターフェイスボードを使う設定は次のようになります。

◉ Panasonic製MO装置付属のASPIマネージャの設定例

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
LASTDRIVE=Z
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥SCSI¥ASPINEC.SYS
DEVICE=A:¥SCSI¥MSDRVR.SYS -olk -va -ot3 -th -i4:va
DEVICE=A:¥DOS¥RAMDISK.SYS 6144
```

ASPINEC.SYSがASPIマネージャで、MSDRVR.SYSがMO装置用のドライバです。

最初に使用したASPIマネージャのディスクに、CD-ROM装置用のドライバやMO装置用のドライバが含まれていれば、あとからそれらの装置を購入してもドライバ類は必要ないはずです。各装置に付属のドライバ類を組み込む前に、使用中のASPIマネージャにセットになっているドライバがあれば、そちらで試すようにしてください。また、あとから装置を付け足す場合、ASPIマネージャを重複して組み込まないように注意しましょう。

PDを使用する2つの方法

PDは、**相変化型光ディスク**と呼ばれる最も新しい記憶装置の1つです。

MOと似ていますが、容量が650Mバイトと大きい上に、CD-ROM装置と兼用で使用できるというメリットがあります。また、ファイルを上書きコピーする場合、MOは、まず以前のファイルを削除してから書き込みを行うのに対して、PDは上書きコピーができるため、書き込み速度が向上しています。おそらく、MO以上のテンポで普及が進んでゆくのではないでしょうか。

なお、MOはMagneto（磁気）とOptical（光）から付けられた名称ですが、PDの開発元（松下電器産業）によると、PDは何かの略ではないそうです。

PDの使用方法に関しては、まさにMO+CD-ROMという感じです。MOとCD-ROMの解説も参考にしてください。MOと同様に、デバイスドライバを使って認識させる方法と、デバイスドライバを使わない方法とがあります。

●ドライバを組み込んで PD を使う

デバイスドライバを用いると、PD をハードディスクのようにパーティションに分けて使用できます（最大 16 分割）。この場合は、SCSI ID を連続状態にならないように設定してください。あとは、付属のインストールプログラムによって、ドライバのインストールと CONFIG.SYS や AUTOEXEC.BAT の書き換えが行われるはずです。

●ドライバなしで PD を使う

次の 2 つの条件がそろえば、PD はデバイスドライバを使わなくても MO として認識されます。

①使用している SCSI インターフェイスボードが PC-9801-55、PC-9801-92 などの MO 装置に対応したものであること。
② MS-DOS Ver.5 以降を使用していること。

ただし、PC-9801-100 や、PC-9821X-B02 などの SCSI ボードでは、どのように SCSI ID を設定しても MO として認識されてしまいます。

ASPI マネージャとデバイスドライバで PD 装置を使用する

PD は、たいてい ASPI マネージャを用いて SCSI カードを認識させてから、デバイスドライバを組み込むようになっているはずです。筆者の使用している LF-1000JD（Panasonic）では、次のような設定になります。

◉ドライバを組み込んで PD を使う設定

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
LASTDRIVE=Z
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥DOS¥RAMDISK.SYS 6144
DEVICE=A:¥SCSI¥ASPINEC.SYS
DEVICE=A:¥SCSI¥MSDRVR.SYS -O1K -VA -OT3 -OB -TH-
DEVICE=A:¥SCSI¥MCDPD.SYS /D:PD_CD /T3
```

ASPINEC.SYS が ASPI マネージャです。もし、ほかの ASPI マネージャが組み込まれていれば、そちらを使うことができます。重複しないように注意してください。

MSDRVR.SYS は、PD にドライブを割り当てるためのドライバです。これにより、複数のパーティション（最大 16）を使用することができます。

MCDPD.SYS は、CD-ROM 用のドライバです。AUTOEXEC.BAT には、MSCDEX.EXE が記述されています。

もしも、CD-ROM 装置が組み込み済みのシステムで、PD 装置では PD のみを使用するつもりなら、CD-ROM ドライバは組み込み済みのものを使ってください。MCDPD.SYS はコメントにしておきます。

デバイスドライバを用いずに PD を使用する

PD を 1 つのパーティションで使用するなら、デバイスドライバを使わないほうが扱いが簡単です（MO がサポートされている SCSI ボードのみ可能）。SCSI ID を連続させておいて、CD-ROM に関するドライバだけを組み込んで使用します。

◉ MSDRVR.SYS の行をコメントにし、MO として PD を使う設定

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
LASTDRIVE=Z
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥DOS¥RAMDISK.SYS 6144
DEVICE=A:¥SCSI¥ASPINEC.SYS
REM DEVICE=A:¥SCSI¥MSDRVR.SYS -01K -VA -OT3 -OB -TH-
DEVICE=A:¥SCSI¥MCDPD.SYS /D:PD_CD /T3
```

LF-1000JD では MSDRVR.SYS を組み込んでおかないとフォーマットプログラム **RXFORM.EXE** が実行できませんが、パーティションを設定しなくてもよければ、**MOFORM98.EXE** というフォーマットプログラムのほうが利用できます。

デバイスドライバを使わないと、PDのメディアをセットした状態で起動すると、PDのほうにシステムを読み込みにいってしまうので注意してください。

●PDをパーティション分割する

PDを複数のパーティションに分ける場合には、MSDRVR.SYSのオプションで必要なドライブ数を確保しておく必要があります。たとえば、3つのパーティションに区切って使用するなら、次のように設定します。

```
DEVICEHIGH=A:¥SCSI¥MSDRVR.SYS -O1K -VA -OT3 -OB -TH- I4:D3
```

「I4:D3」の部分のうち、I4はSCSI IDが4、D3は3つのドライブ名を確保するという意味になります。

PC-9801-100またはPC-9821X-B02で使用する場合の注意

SCSIボードとしてPC-9801-100またはPC-9821X-B02を使用している場合には、ASPIマネージャに-M1スイッチが必要になります。

```
DEVICE=A:¥SCSI¥ASPINEC.SYS -M1
```

PC-9801-100やPC-9821X-B02では、MSDRVR.SYSを組み込むとPDに2つのドライブ（CD-ROMの分を除く）が重複して割り当てられてしまいます。どちらのドライブも、内容は同じになるのですが、PDのマニュアルを読む限りでは、そのような使い方はしないほうがよさそうです。

マニュアルには、フォーマットのときにだけMSDRVR.SYSを組み込むという記述もありますが、MOFORM98.EXEを利用すればMSDRVR.SYSは必要ありません。ただし、MOFORM98.EXEではパーティションの設定はできなくなっています。

結局のところ、PC-9801-100やPC-9821X-B02ではMSDRVR.SYSを用いず、パーティションも切らずに利用するということになると思います。

※本書の執筆時点では、PDに関してはほとんど情報が入ってきていません。そこで、追加情報などがあれば、本書の付属ディスクのREADME.DOCに入れておくことにしますので、内容を確認してください。

Step 37

サウンドボード「SOUND BLASTER 16」を使用する

●ポイント●

・サウンドボード用のドライバは、ボード上のI/Oアドレス、割り込みレベル（IRQ）の設定に合わせる。
・Windows環境では、DOS用のドライバは必要ないが、AUTOEXEC.BAT内の環境変数は設定が必要。
・サウンドボードに接続するCD-ROMは、CONFIG.SYSでドライバを組み込んで、AUTOEXEC.BATでMSCDEX.EXEを常駐させる。

●解　説●

サウンドボードの種類

PC-9800シリーズには、機種によって搭載されている音源の種類が異なります。旧タイプの機種では**FM音源**が用いられていましたが、最近はWindows環境でサウンドを鳴らすための**PCM音源**が主流となっています。

しかし、サウンドを重視するゲームソフトでは、内蔵されている音源だけでは十分な効果が得られないということで、MIDI装置や市販されている特定のサウンドボードに対応しているものも多くなっています。

ここでは、CREATIVE LABS社の「**SOUND BLASTER 16**」という16ビットサウンドボードを例にとって、CONFIG.SYSやAUTOEXEC.BATでの記述方法について解説します。

SOUND BLASTER 16の組み込み

SOUND BLASTER 16は、ボード上のジャンパを正しく設定してから、本体に差し込むようにしてください。特に、I/Oアドレスと割り込みレベル(IRQ)の設定は重要です。これらが、ほかのボード類と重なっていると、ボードを組み込んでも音が出ないことになるからです。

サウンドボード用のデバイスドライバに対しても、I/Oアドレス、割り込みレベル、DMAチャネルの設定は行いますが、それらは、あくまでボード上の設定に合わせるというだけで、メインはボード上の設定なのです。

PC-9800シリーズの場合、システムの拡張具合によって環境が大きく異なるDOS/V機とは違い、本来なら出荷時の設定でもたいていのシステムでは問題が起きないはずなのです。しかし最近はTVチューナーボード、MPEG再生ボードというように、使われる拡張ボードが急激に増えているため、割り込みレベルの競合という問題も十分考えられるようになってきています。せっかく購入しても、割り込みレベルなどがいっぱいでは使うことができないので、複数のボード類を用いているシステムではあらかじめ確認しておくほうがよいでしょう。

●インストール時のCONFIG.SYSの変更点

ボードを差し込んだら、DOSを起動して、付属のディスクからINSTALL.EXEを実行してください。プログラムのインストールと、CONFIG.SYSおよびAUTOEXEC.BATの書き換えが行われます。

インストール後のCONFIG.SYSには、次のような記述が加えられているはずです。

◉インストールプログラムでCONFIG.SYSに加えられた記述

```
DEVICE=C:¥SB16¥DRV¥CTSB16J.SYS /UNIT=0 /BLASTER=A:D2 I:10 D:3
DEVICE=C:¥SB16¥DRV¥CTMMSYS.SYS
DEVICE=C:¥SB16¥DRV¥CSP.SYS /UNIT=0 /BLASTER=A:D2
```

CTSB16J.SYS にあるオプションのうち、/BLASTER スイッチによって、次の指定を行います（＊印は初期値)。

◉ CTSB16J.SYS の/BLASTER スイッチの設定

スイッチ	意味	値
A:	サウンドボードの I/O アドレス	＊ D2,D4,D6,D8,DA,DC,DE
I:	割り込みレベル	3,＊ 5,10,12
D:	8 ビットまたは 16 ビットデータ転送用の DMA チャネル	0,＊ 3

組み込み時に、各設定がほかのボード類と重複していないかどうかのテストが行われますが、絶対というわけではないため、不具合が生じた場合には、あやしいと思われる設定を変更してみてください。もちろん、ボード上のジャンパも変更します。

CSP.SYS は、ボードにアドバンストシグナルプロセッサが付加されている場合のみに組み込まれます。

●インストール時の AUTOEXEC.BAT の変更点

AUTOEXEC.BAT には、次の記述が加えられます。

◉インストールプログラムで AUTOEXEC.BAT に加えられた記述

```
SET SOUND=A:¥SB16
SET BLASTER=Ad2 I10 D3
SET MIDI=SYNTH:1 MAP:E
A:¥SB16¥DIAGNOSE /S
A:¥SB16¥SB16SET /M:200 /VOC:200 /CD:200 /MIDI:200 /LINE:200 /TREBLE:0 /Q
```

環境変数 **BLASTER** の内容は、CTSB16J.SYS のオプションに対応します。この設定に基づいて、**DIAGNOSE.EXE** によってハードウェアの設定を行っています。

SB16SET コマンドの各スイッチでは、サウンドのボリュームを設定しています。

◉ SB16SET コマンドのスイッチ

スイッチ	意　味
/M:l,r	マスター出力ボリューム
/VOC:l,r	ボイス出力ボリューム
/CD:l,r	CD 入力ボリューム
/MIDI:l,r	MIDI 入力ボリューム
/LINE:l,r	ライン入力ボリューム
/TREBLE:l,r	高音コントロール
/BASS:l,r	低音コントロール
/SPK:l,r	スピーカー出力ボリューム
/MIC:n	マイク入力ボリューム

l（左）、r（右）、n には、0～255 までの範囲でボリュームを設定してください。

左右の音量が同じ場合には、

```
/M:200
```

のように指定し、左右の音量が異なる場合には、

```
/M:100,150
```

のように指定します。

/Q スイッチを付けると、実行時にメッセージを表示しなくなります。

SOUND BLASTER 16 には、そのままでも MIDI ファイルを再生する機能がありますが、音色の数が少ないため、やはり本格的に MIDI を楽しみたいなら MIDI 音源装置の購入をお勧めします。

Windows 環境のための設定

CONFIG.SYS で組み込まれるサウンドドライバは、DOS 環境のためのものです。したがって、Windwos 環境だけで利用するのであれば、すべて削除してしまってもかまいません。

ただし、AUTOEXEC.BAT の環境変数 (SOUND、BLASTER、MIDI) は設定しておいてください。

CD-ROM 接続時はインターフェイスに注意

SOUND BLASTER 16 には、CD-ROM を接続するための専用タイプのインターフェイスが付いています。専用タイプのインターフェイスには決められた CD-ROM 装置しか接続できないので注意してください。

CD-ROM を接続した場合の設定は、CONFIG.SYS で付属のドライバを組み込み、AUTOEXEC.BAT で MSCDEX.EXE を組み込みます。「Step36 SCSI インターフェイスで CD-ROM 装置／MO 装置／PD 装置を使用する」の解説を参考にしてください。

Step 38

ノートパソコンの設定「PC カード」(PCMCIA カード) を使う

●ポイント●

- PC カードとは、PCMCIA と JEIDA との間で規格の標準化を進めているメモリカードや I/O カードのこと。
- PC カードでは、基本的にパソコンの本体に付属になっているドライバを用いて、それらのドライバが未対応のカードでは、カードに付属のドライバを用いる。
- インストールされているすべてのカード用ドライバが必要とは限らない。
- PC カードが使用するメモリ (資源) の範囲は、カードの種類によっては追加が必要。

●解 説●

PC カード／PCMCIA カードとは

PC カードは、主にノート型パソコンなどで使用されているクレジットカードサイズの拡張機器です。最近では、デスクトップタイプのパソコンでも利用されるようになっています。

もともとは、**PCMCIA** (Personal Computer Memory Card International Association：カードのハードウェアやソフトウェアの規格をとりまとめている機関) によって規格が決められたため、今のところ **PCMCIA カード**と呼ばれることのほうが多いかもしれません。PC カードという名称は、**JEIDA (社団法人日本電子工業振興協会)** が PCMCIA と共同で標準化を進めているカードのことをいいます。

PC カードには、3 つのタイプがあります。

◉ PC カードの種類

タイプ名	厚さ
タイプⅠ	3.3mm
タイプⅡ	5.0mm
タイプⅢ	10.5mm

タイプⅢは、タイプⅡの2倍の厚みがあります。そのため、通常はタイプⅢを差し込むスロットに、タイプⅡのカードが2枚差し込めるようになっています。

現在、PC カードには、次のような種類があります。

- メモリカード：フラッシュメモリカード、SRAM カード、ROM カードなど
- モデムカード／FAX モデムカード
- LAN（ネットワーク）カード
- ハードディスク(ATA) カード
- サウンドカード
- SCSI カード

※モデムカード／FAX モデムカード、LAN カード、ハードディスクカード、サウンドカード、SCSI カードを総称して、I/O カードと呼びます。

PC カードドライバの設定方法

PC カードは、Plug&Play という規格に対応しています。Plug&Play とは、面倒な設定をしなくても、拡張カード（ボード）を差し込むだけで、すぐに使えるようになるという意味で、従来のように、割り込みレベル(IRQ)や I/O アドレスなどで悩むことはないのです。

PC カードのためのドライバソフトは、カードを購入すると付属しているはずですが、ノート型パソコンなどでは、あらかじめインストールされている場合があります。どちらを用いたらよいかということは、購入したカードのマニュアルを読んで判断してください。

基本的にはパソコンに付属のドライバを用いて、正しく認識されない場合だけ、カードに付属のドライバで補うという使い方になると思います。

本書では、PCカードのドライバソフトがインストールされている、NEC PC-9821 Lt/540Aという機種を例にとって解説します。

PC-9821 LtのCONFIG.SYSとAUTOEXEC.BAT

PC-9821 Lt/540Aには、あらかじめDOS6とWindwosがインストールされています。そして、次のようなCONFIG.SYSが作成されています。

◉ PC-9821 LtのCONFIG.SYS

```
FILES=40
BUFFERS=10
SHELL=A:¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB /E=DC00-DFFF
DEVICE=A:¥DOS¥SETVER.EXE
DEVICE=A:¥DOS¥SSDRV.SYS————————————————①
DEVICE=A:¥DOS¥CS.EXE———————————————————②
DEVICE=A:¥DOS¥CSALLOC.EXE A:¥DOS¥CSALLOC.INI———③
DEVICE=A:¥DOS¥CDCOMP.SYS———————————————④
DEVICE=A:¥DOS¥MCDRV.SYS————————————————⑤
DEVICE=A:¥DOS¥HDCARD.SYS———————————————⑥
DEVICE=A:¥DOS¥CARDID.EXE A:¥DOS¥CARDID.INI—————⑦
DOS=HIGH,UMB
```

※ DEVICEHIGHコマンドの部分は、すべてDEVICEコマンドに変えてあります。

PCカード用のドライバは1〜7です。それぞれ、次のような役割があります。

① SSDRV.SYS（ソケットサービス）

ハードウェアを操作するためのドライバで、機種によって専用のドライバが必要となります。カードソケットの初期化を行い、ソケットおよびソケットに挿入されたカードにアクセスするための共通のインターフェイスを提供します。

② CS.EXE（カードサービス）

カードの挿入や抜き取り、バッテリー状態の変化などを通知するほか、ソケット、メモリウィンドウ、I/Oウィンドウ、割り込みレベルなどのリソース(資源）を集中管理します。

③ CSALLOC.EXE（資源確保ユーティリティ）

システムの起動時に、カード以外の機器によって使用されているリソースを見つけだし、カードサービスがリソースを重複して使用しないようにします。

コマンドとして実行すると、資源に関する設定を行い、結果をCSALLOC.INIファイルに書き込むことができます。

このデバイスドライバは、1度実行されるだけでメモリには常駐しません。

④ CDCOMP.SYS（従来互換ドライバ）

PC-9801NX/C,P,NS/A,NL/R,PC-9821Neで提供されたカードサポートソフトウェア上で動作するカードドライバをLtでも動作させます。

⑤ MCDRV.SYS（メモリカードドライバ）

フラッシュメモリカード、SRAMカード、ROMカードなどを使うために、スロットに挿入したカードにドライブを割り当てます。

⑥ HDCARD.SYS（ATAカードドライバ）

JEIDA/PCMCIA仕様に準拠しているハードディスク(ATA)カードを、通常のハードディスクと同様に使えるようにします。

⑦ CARDID.EXE（カード自動設定ドライバ）

カードの抜き差しを検出し、挿入されたカードの種類を自動的に判断して、カードとソケット／アダプタの設定を行います。

それぞれの役割について、理解しておく必要は特にありません。

これらのドライバのうち、必ず組み込まなければならないのは、SSDRV.SYS、CS.EXE、CSALLOC.EXEです（Windows環境でも必要です)。

CDCOMP.SYSは、ほとんど必要ないでしょうし、MCDRV.SYS、HDCARD.SYSもメモリカードやハードディスクカードを使わなければ不要です。

CARDID.EXEは、絶対に必要というドライバではありませんが、たいていのカードを自動認識してくれますし、Windowsのカードユーティリティを使うためにも組み込んでおくほうがよいと思います。

使わないドライバの行は、REMコマンドでコメントにしておきましょう。PCカードをまったく使わないのであれば、①〜⑦の行をすべてコメントにしておきます。

EMM386.EXEに/Eスイッチが付けられているのは、カードが使用するメモリをUMBメモリから除外するための指定です。

SCSIカードを使う

ここでは、キャラベルデータシステムの『**PSC-SCSI**』というSCSIインターフェイスカードを用いて、CD-ROM装置とMO装置を接続した場合の設定例をお見せします。

SCSIカードに付属のソフトは、SCSICARDというディレクトリにインストールしてあるものとします。

◉ PSC-SCSI使用時のCONFIG.SYS

```
FILES=40
BUFFERS=3
LASTDRIVE=Z
SHELL=A:¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB /E=DC00-DFFF
DEVICE=A:¥DOS¥SETVER.EXE
DEVICE=A:¥DOS¥SSDRV.SYS
DEVICE=A:¥DOS¥CS.EXE
DEVICE=A:¥DOS¥CSALLOC.EXE A:¥DOS¥CSALLOC.INI
REM DEVICE=A:¥DOS¥CDCOMP.SYS
REM DEVICE=A:¥DOS¥MCDRV.SYS
REM DEVICE=A:¥DOS¥HDCARD.SYS
DEVICE=A:¥DOS¥CARDID.EXE A:¥DOS¥CARDID.INI
DOS=HIGH,UMB
DEVICE=A:¥SCSICARD¥MPS98_CS.SYS
DEVICE=A:¥SCSICARD¥MPSBIOSW.SYS
DEVICE=A:¥SCSICARD¥MPSDRIVE.SYS
DEVICE=A:¥SCSICARD¥CDSCD.SYS /D:CDROM
```

BUFFERSの値は、3以下にしないとHMAメモリからはみ出てしまいます。

CD-ROM装置などを接続するので、LASTDRIVEコマンドでドライブ数を増やしてあります。

下のほうにある、**MPS98_CS.SYS**と**MPSBIOSW.SYS**がSCSIカードのためのドライバです。**MPSDRIVE.SYS**はMO装置を使うためのドライバ、**CDSCD.SYS**がCD-ROM装置を使うためのドライバになっています。CD-ROM装置を使うなら、AUTOEXEC.BATでMSCDEX.EXEも実行します。

なお、単独でSCSI装置を使う場合、SCSI装置が認識されないというトラブルが発生するかもしれません。筆者もMO装置で経験したのですが、これを解決するためにはほかのSCSI装置を一緒に接続してみてください。たとえば、ハードディスクやCD-ROM装置をSCSI接続に加えるのです。困ったことですが、今のところほかに解決策があるか不明です。

カード資源の変更

LANカードなどは、カードが使用する**メモリウィンドウ**のサイズが特に大きくなっています。もしもメモリウィンドウの不足が生じたら、設定を変更してメモリの拡張を行ってください。

まず、EMM386.EXEの/Eスイッチで指定する領域を加えます。初期設定でDC000h～DFFFhの16Kバイトが確保されていますが、このうち8Kバイトはカードサービスが使用するため、メモリウィンドウとしては8Kバイトが確保されていることになります。

たとえば、もう8Kバイト増やすなら、CONFIG.SYSに

```
DEVICE=A:¥DOS¥EMM386.EXE /UMB /E=DC00-DFFF /E=D600-D7FF
```

と設定してください。16Kバイトなら/E = D400-D7FFです(必ず4Kバイト単位で指定)。

次に、**CONFIG**コマンドで**資源**の設定を行います。

①次のように入力して CONFIG コマンドを起動します。

```
A:¥>CONFIG
```

②カード環境設定ユーティリティが起動したら、GRPH + U キーを押して「ユーティリティ」のメニューを出し、「資源の設定」を実行してください。
③ GRPH + O キーの「開く」を実行し、CSALLOC.INI を読み込みます。
④ TAB キーを押して画面の下にある「変更した資源」のなかの「メモリ」にカーソルを移動して、GRPH + E キーを押してください。
⑤修正メニューが表示されるので、GRPH + A キーで「追加」を実行します。
⑥メモリに表示されている C000 の部分を追加したい先頭アドレスに、EFFF の部分を最終アドレスにそれぞれ変更してください。たとえば、D6000h ～D7FFFh なら、C000 を D600 に、EFFF を D7FF に変更します。
⑦入力が終わったら、TAB キーを押して「OK」にカーソルを移動してから、⏎ キーを押してください。確認の画面が出るので、再び「OK」で ⏎ キーを押します。
⑧元の設定画面に戻るので、TAB キーでカーソルを「OK」に移動して、⏎ キーを押してください。GRPH + S キーで CSALLOC.INI に設定内容が保存されます。

これで設定は完了です。ESC キーを押してから、GRPH + F キーを押して、メニューにある「終了」を実行してください。

Step 39

DoubleSpace を使う

●ポイント●

- DoubleSpace を使うには、DBLSPACE コマンドで圧縮を実行するだけでよい。
- 圧縮ドライブを作成すると、CONFIG.SYS で DBLSPACE.SYS が/MOVE スイッチ付きで組み込まれる。
- DoubleSpace の組み込み時には、BUFFERS の値を 1～3 程度まで減らすとメモリの節約になる。
- 圧縮を解除するには、十分な空き容量が必要になる。
- DoubleSpace は、起動ドライブのルートディレクトリに DBLSPACE.BIN と DBLSPACE.INI の 2 つのファイルがあると自動的に組み込まれる。
- 圧縮ボリュームファイル、DBLSPACE.BIN、DBLSPACE.INI は、いずれもシステム属性、隠し属性、リードオンリー属性がセットされている。

●解 説●

DoubleSpace 組み込み時の設定

DoubleSpace は**ディスク圧縮ソフト**と呼ばれるユーティリティです。ディスクの中身を圧縮して、2 倍近くに広げてくれるありがたいソフトですが、どうしてもディスクのアクセス速度が低下してしまうのと、圧縮後のドライブ名は新たに決められるため、ドライブ数が増えて面倒になるといった問題が生じます。

何より、圧縮ドライブに万一のことがあると、中のファイルを復活させることが非常に困難であることから、筆者としてはあまりお勧めできません。現在では、ハードディスクの価格が驚くほど低下していますから、大きなリスクを背負ってまでディスク圧縮ソフトを使う必要はないと思うのです。

それでも、ノートパソコンなどでどうしてもディスク容量が不足している場合には、使うのもやむをえないでしょう。ここでは、DoubleSpace の具体的な使い方に関しては説明を省き、CONFIG.SYS に関係のあることだけを取り上げることにします。

● DBLSPACE に UMB と HMA を使わせるのがポイント

DoubleSpace は、**DBLSPACE** コマンドを実行して、圧縮ドライブの作成などを行います。DoubleSpace は、システムによって自動的に認識されてメモリに読み込まれるため、CONFIG.SYS などでドライバとして組み込む必要はありません。ただし、圧縮ドライブを作成すると、CONFIG.SYS には次の 1 行が加えられているはずです。

◉圧縮ドライブ作成時に CONFIG.SYS に加えられる記述

```
DEVICEHIGH=A:¥DOS¥DBLSPACE.SYS /MOVE
```

これは、DoubleSpace のプログラムを UMB メモリに常駐させるための設定です。この設定がなかったり、UMB メモリに十分な空きがなければ、コンベンショナルメモリ（基本メモリ）に常駐します。

DoubleSpace の常駐プログラムは、HMA メモリを利用すると常駐サイズが小さくなります。そこで、BUFFERS の値をさらに減らして、ディスクバッファと DoubleSpace の両方が入るように調整するとよいでしょう。

BUFFERS に指定すべき値は機種により異なります。たとえば、PC-9821 Xa/C10W では 3 以下にしないとすべてが収まりませんでしたし、PC-9821 Lt/540A では 1 にまで減らさないとだめでした。設定を少しずつ変更しながら再起動し、MEM コマンドを実行して空きメモリを確認することにより、最適な値を見つけてください。

圧縮ドライブの強制的な削除

実は、DoubleSpaceによる圧縮ドライブは、作るよりも消すほうがたいへんなのです。基本的には、DBLSPACEコマンドを実行して、「ツール」のメニューにある「圧縮解除」を実行すればよいのですが、そのためには圧縮ドライブを作成したドライブに十分な空きが必要となります。空き容量が不足しているからこそ、ディスクを圧縮したはずですから、それ以外に空きを確保しておけというのはなかなか無理な注文でしょう。結局、強制的に圧縮ドライブを削除しなければならないケースが多くなるはずです。

●通常の圧縮ドライブの削除

圧縮ドライブは、システム上はDBLSPACE.000からのファイル名を付けられた圧縮ボリュームファイルとして管理されています。圧縮ボリュームファイルの削除は、やはりDBLSPACEコマンドによって行います（削除を実行する前には、当然のことながら大事なファイルをほかのドライブに移動しておかなければなりません）。メニューの「ドライブ」にある「削除」を実行してください。これにより、圧縮ボリュームファイルとなっていた圧縮ドライブは削除されます。

●起動ドライブ全体が圧縮されているときの削除

起動ドライブ全体を圧縮していた場合には、「削除」を実行することはできませんから、圧縮ボリュームファイル（DBLSPACE.000～）を手作業で削除してください。ただし、システム属性、隠し属性、リードオンリー属性がセットされていますから、次のようにして属性の変更を先に行う必要があります（DBLSPACE.000の場合）。

◉圧縮ボリュームファイルの属性変更

```
A:¥>ATTRIB -R -S -H DBLSPACE.000
A:¥>DEL DBLSPACE.000
```

すべての圧縮ボリュームファイルを削除して、完全にDoubleSpaceを使わないのであれば、ほかにも作業が必要です。圧縮ボリュームファイルを削除しただけの状態では、相変わらずDoubleSpaceがメモリに常駐してしまうからです。

まず、CONFIG.SYS にある次の1行を削除してください。

```
DEVICEHIGH=A:¥DOS¥DBLSPACE.SYS /MOVE
```

DoubleSpace は、起動ドライブのルートディレクトリに **DBLSPACE.BIN** と **DBLSPACE.INI** の2つのファイルがあると、自動的に組み込まれるようになっています。

それらのファイルは、圧縮ボリュームファイルと同様に、システム属性、隠し属性、リードオンリー属性がセットされていますから、単に DIR コマンドを実行しただけでは見ることができません。確認するには、次のように実行してみてください。

```
A:¥>DIR /A:H
```

それらを削除するには、次のように実行します。

```
A:¥>ATTRIB -R -S -H DBLSPACE.BIN
A:¥>ATTRIB -R -S -H DBLSPACE.INI
A:¥>DEL DBLSPACE.BIN
A:¥>DEL DBLSPACE.INI
```

DBLSPACE.INI だけを削除しても、常駐することはなくなりますが、少しでもディスク容量を節約したければ、DBLSPACE.BIN も一緒に削除してしまうほうがよいでしょう。DBLSPACE.BIN は、DOS6 をインストールしたディレクトリに通常ファイルとして存在していて、圧縮ドライブの作成と共に、属性を変えてルートディレクトリにコピーされています。ですから削除してしまっても、次に圧縮ドライブを作成する際に問題になることはないのです。

Step 40

DPMI 環境のための設定「一太郎 Ver.5」を使う

●ポイント●

- 一太郎 Ver.5 などのジャストシステム製品は、DPMI 環境で動作する。
- DOS6 で DPMI サーバを組み込むには、EMM386.EXE に/DPMI スイッチを指定しておいてから、DPMI.EXE を実行する。
- DPMI.EXE は、デバイスドライバとして組み込むこともできるが、あとから常駐を解除することができなくなる。

●解 説●

DPMI サーバの起動方法

DPMI（DOS Protected Mode Interface）というのは、Intel 社が中心となって作成した拡張メモリ仕様で、現在の Windows は、この DPMI 環境で動作しています。DOS から起動するアプリケーションのなかにも、いくつか DPMI に対応しているものがありますが、その代表は『一太郎 Ver.5』（DOS 版）をはじめとするジャストシステム製品と言えるでしょう。

DPMI では、DPMI 環境を提供するドライバのことを **DPMI サーバ**、DPMI 環境で動作するアプリケーションのことを **DPMI クライアント**と呼んでいます。DOS6 の拡張メモリマネージャ EMM386.EXE には、DPMI に対応させるためのオプションがあって、DPMI サーバとして DPMI.EXE が付属しています。一太郎 Ver.5 などにも、拡張メモリマネージャが付属してはいますが、EMM386.EXE に比べると機能的に劣っているため、ここでは DOS6 に付属のドライバを用いた設定方法について解説しておきます。

● DOS6のドライバでDPMIを有効にする設定

DPMIを有効にするためには、次のようなCONFIG.SYSになります。

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB /DPMI
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥DOS¥RAMDISK.SYS 2048
```

EMM386.EXEに/DPMIスイッチを付けるだけです。

ただし、EMSメモリが少なくとも16ページ（256Kバイト）必要になるので、特にEMSメモリのサイズを指定している場合には注意してください。

● DPMIコマンドの実行

こうしたあと、AUTOEXEC.BAT内で**DPMI**コマンドを記述します。

```
@ECHO OFF
PATH=A:¥;A:¥DOS
SMARTDRV 3072
SET DOSDIR=A:¥DOS
SET TEMP=（作業用ドライブ名、ディレクトリ名）
PROMPT $P$G
DPMI
```

これで、DPMIサーバが起動した状態になるので、一太郎Ver.5などのDPMIクライアントが利用できます。この状態でも、通常のDOSアプリケーションの起動にさしつかえはありません。

● DPMIサーバの常駐解除

DPMIサーバの常駐を解除するには、/Rスイッチを付けて実行してください。

```
A:¥>DPMI /R
```

● DPMI サーバをドライバとして組み込む方法

DPMI.EXE は、コマンドとして実行するかわりに、デバイスドライバとして組み込むこともできます。

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB /DPMI
DEVICE=A:¥DOS¥DPMI.EXE
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥DOS¥RAMDISK.SYS 2048
```

ただし、こうするとあとから常駐を解除することはできなくなります。

Step 41

Windowsを起動するための設定

●ポイント●

- 十分な拡張メモリが搭載されていれば、Windows を起動するために特別な設定は必要ない。
- Windows 専用の設定にするなら、日本語 FEP などの不要なドライバは外しておく。
- EMS メモリを使うプログラムがなければ、EMM386.EXE に/NOEMS スイッチを付けて、UMB メモリを増やすことができる。
- CD-ROM 用のドライバや、PC カード用のドライバなどは Windows でも必要。
- WX III for Windows を DOS ボックスで利用するには、専用のドライバを組み込んでおく。
- VJE-Delta は、そのままで DOS ボックスでも利用できる。

●解 説●

Windowsのために特別な設定は必要ない

最近では、Windows がインストールされた状態で出荷されている機種も増えてきましたが、それらで使われている CONFIG.SYS は、DOS のアプリケーション用、Windows 用というようには分けられてはいないはずです。今や、十分な拡張メモリさえ搭載されていれば、特別な設定をしなくても Windows を起動させることができるからです。

ただし、DOSの環境がまったくWindowsに影響しないというわけではありません。いくら拡張メモリが豊富にあっても、コンベンショナルメモリ（基本メモリ)が不足してしまうと、Windowsのアプリケーションが起動できなくなってしまうのです。

ですから、DOS環境とWindows環境を共通のCONFIG.SYSで起動させようと思うなら、できる限りコンベンショナルメモリが空くように設定してください。不要なドライバ類は組み込まないように、なるべくUMBメモリを活用するように心掛けましょう。

なお、FILESの値は必ず30以上にしてください。

Windows専用の設定をする

もしも、DOSのアプリケーションを使うことなく、最初から最後までWindows環境で利用するなら、やはり専用の設定をするほうがよいでしょう。

Windows環境では、DOS環境で組み込んでいた次のドライバ類が必要ありません。

◉ Windows環境では必要ないドライバ

- プリンタドライバ
- マウスドライバ
- RS-232Cインターフェイスドライバ
- 日本語FEPドライバ（補助ドライバを含む）
- サウンドボード関連のドライバ（環境変数の設定は必要）

また、RAMディスクやディスクキャッシュに割り当てる拡張メモリは少なめにして、できる限りWindows用に残しておくようにしてください。10Mバイト程度の拡張メモリがあると、かなり快適に使用できるはずです。

Windows専用のCONFIG.SYSの例をお見せします。

◉ Windows専用のCONFIG.SYSの例

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥DOS¥RAMDISK.SYS 2048
```

このように、たいていはごくシンプルな内容になるはずです。

なお、EMSメモリを使うプログラムを組み込まないのであれば、EMM386.EXEの/UMBスイッチを/NOEMSスイッチに変えておくと、EMSメモリが利用できなくなるかわりに、UMBメモリが64Kバイト増えます。

◉/NOEMSスイッチでEMSページフレームをUMBにする

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /NOEMS
DEVICE=A:¥DOS¥RAMDISK.SYS 2048
```

このほか、CD-ROM用のドライバや、PCカード用のドライバなどは組み込むようにしてください。

SETVER.EXEは、バージョンテーブルを設定している場合だけ組み込みます。

DOS ボックスで利用できる FEP を組み込む

最新の日本語 FEP のなかには、Windows だけではなく、**DOS ボックス**(Windows 内で起動した DOS で、DOS 窓とも呼ばれる)で日本語入力が可能なものがあります。これから、それらの設定方法について解説します。

● WX III for Windows

『**WX III for Windows**』では、バージョン 3.03 から DOS ボックスに対応するようになりました。通常、Windows 対応の FEP を CONFIG.SYS で組み込むことはありませんが、この場合には DOS ボックス専用のドライバが必要になります。

◉ DOS ボックス専用ドライバの組み込み

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥WX3WIN¥WXKW98.SYS
DEVICE=A:¥WX2WIN¥WX3W98.SYS /A5 /INI=A:¥WX3WIN¥WX3SYS98.INI
DEVICE=A:¥DOS¥RAMDISK.SYS 2048
```

WXKW98.SYS、**WX3W98.SYS** の 2 つが、DOS ボックス用のドライバです。それらの前には、必ず KKCFUNC.SYS を記述しておきます。

なお、DOS ボックス用のドライバと、DOS 用の『WX II＋ Ver.2.70』とを同時に組み込むことは可能ですが、DOS 用の『WX III』と一緒に組み込むことはできません。

● VJE-Delta

『**VJE-Delta**』では、DOS ボックスで日本語入力を行うために特別なドライバは必要ありません。Windows で VJE-Delta を使っていれば、そのままで使用することができます。

なお、DOS 環境で利用する設定方法については、「Step34 市販の日本語 FEP を使用する」の「VJE-Delta を組み込む」で解説しています。

Part 3
応用編

応用編の内容

この応用編では、本書の最大のテーマである「メモリの最適化」を扱っています。

まずはじめに、PC-9800シリーズのメモリ構成や、メモリに関する用語について学んでいただくことになります。これは、「DOSは今回のバージョン6.2が最初」という方々への配慮でもありますが、応用編や、次の「Part4 Other Tools編」で解説するテクニックを使いこなすためには、どうしても知っておいていただかないと困る内容なのです。

従来のDOSでメモリの最適化を行うには、ある程度の知識と、試行錯誤を繰り返す根気が必要だったと思います。しかし、DOS6には「MemMaker」というメモリ最適化のためのユーティリティが付属になっているため、何の苦労もなく自動的に最適化が行われてしまうようになりました。

「MemMaker」が登場したことにより、こだわり派が楽しむ余地はなくなってしまったのでしょうか?

いえいえ、ご安心ください。まだいくらでも残されていますから。この応用編から先こそが本書の真骨頂なのです。

Step 42

2進法と16進法

●ポイント●

・コンピュータでは、すべて2進法を用いて計算が行われている。
・2進法は桁数が多くなってしまうため、プログラミングなどでは人間が扱いやすい16進法が用いられている。
・2進法の各桁のことをビットと呼ぶ。
・nビットでは、2のn乗種類のデータを扱うことができる。
・8ビットを1バイトという。
・バイトは、1024倍ごとにキロバイト、メガバイト、ギガバイトとなる。

●解 説●

10進法、2進法、16進法

現在のデジタルコンピュータでは、電圧の高さに2つのレベルを設け、それらに0と1を割り当てて計算を行っています。一般の計算が0～9までの数字を用いる**10進法**であるのに対して、コンピュータ内部で行われている計算を**2進法**と呼びます。

10進法と2進法で、0から15までの数字がどのように表現されるかを見てみましょう。次ページの表のようになります。

◉10 進法と 2 進法の表現の違い

10 進法	2 進法	10 進法	2 進法
0	0	8	1000
1	1	9	1001
2	10	10	1010
3	11	11	1011
4	100	12	1100
5	101	13	1101
6	110	14	1110
7	111	15	1111

2 進法では、10 進法と比較してたいへん桁数が多くなってしまいます。これでは人間が扱うには不便ですから、プログラミングなどでは、2 進法との相互変換が容易で、しかも、桁数が少なくて済む**16進法**がよく使われます。

16 進法では、0〜9 までの数字のほかに、A〜F までの英字を用います。10 進法の 10 が A になり、以下、11＝B、12＝C、13＝D、14＝E、15＝F、という具合に対応します。

◉10 進法、16 進法、2 進法の対応表

10 進法	16 進法	2 進法	10 進法	16 進法	2 進法
0	0	0	11	B	1011
1	1	1	12	C	1100
2	2	10	13	D	1101
3	3	11	14	E	1110
4	4	100	15	F	1111
5	5	101	16	10	10000
6	6	110	17	11	10001
7	7	111	18	12	10010
8	8	1000	19	13	10011
9	9	1001	20	14	10100
10	A	1010	:	:	:

なお、ほかの値と区別できるように、16進法の値を表記する際には、後ろにH(h)を付けることになっています。

ビットとバイト

2進法では、次のように、桁数に応じて表現できる数値の種類が増えていきます（もちろん10進法でも同じことですが）。

◉ 2進法の桁数と表現できる数値の種類

10進表現	2進法1桁	2進法2桁	2進法3桁
0	0	00	000
1	1	01	001
2		10	010
3		11	011
4			100
5			101
6			110
7			111
表現できる種類	2種類	4種類	8種類

どんどん桁数を増やしていくと、n桁の2進数では、「2のn乗」種類の数が表現できることになります。これは、n桁の10進数では「10のn乗」種類の数が表現できるのと同じことです。

2進法では、各桁のことを**ビット**といいます。つまり、**nビット**といえば、「2のn乗種類のデータが扱える」ということになるのです。

ビットは、人間がプログラミングなどで扱う単位としてはちょっと細かすぎます。そこで、8ビットを1つの単位として扱うことになったのです。この単位のことを**バイト**（**Byte**）と呼びます。

メガバイト、キロバイト、ギガバイト

10 進法では、ある単位の 1000 倍になるとキロ、キロの 1000 倍でメガ、メガの 1000 倍でギガというように、大きな単位になっていきます。これは、2 進法にも当てはまりますが、2 進法では 1024（2 の 10 乗）ごとに繰り上がる点が異なっています。

メモリの単位として用いられているバイトは、1024 倍ごとに、**キロバイト**（KB）、**メガバイト**（MB）、**ギガバイト**（GB）になります。

8 ビットが 1 バイトになり、1024 バイトが 1 キロバイトになる、という構造を表にすると次のようになります。

◉ 2 進法の単位

ビット	バイト	K バイト	M バイト	G バイト	バイト数の 10 進表現
8	1				1
	1024	1			1024
		1024	1		1,048,576
			1024	1	1,073,741,824

Step 43

CPU、メモリ、アドレスとは

●ポイント●

- CPU は、パソコンの演算と制御を行う。
- 32 ビット CPU としては、80386、80486、Pentium が用いられている。
- 同じ種類の CPU であれば、クロック数が速いほうが処理速度も上になる。
- メモリはパソコンのデータやプログラムを記憶するための装置。
- メモリには ROM と RAM がある。
- CPU がメモリに対して読み書きを行う際には、場所を特定するためにアドレスを用いる。
- 16 ビットパソコンでは、物理アドレスをそのまま表現できないため、セグメントアドレスとオフセットアドレスを用いる論理アドレス方式が採用されている。

●解　説●

CPUとは

CPU（Central Processing Unit：中央処理装置）とは演算と制御を行うパソコンの頭脳です。

CPUは、与えられたプログラムに従って計算などを行う部分ですから、「パソコンの性能＝CPUの性能」と言えるのです。CPUが一度に扱うことのできるデータ量（ビット数）によって、**～ビットパソコン**と総称されます。

現在、PC-9800シリーズの32ビットパソコンに使われているのは、インテル社の**80386**、**80486**、Pentiumという CPUです。

80386には、廉価版の**80386SX**があり、通常のタイプを**80386DX**と呼んで区別しています。また、携帯用パソコンのために節電機能をもたせた**80386SL**もあります。

80486にも、80386と同様に廉価版の**80486SX**があります。これは、通常のタイプ（＝80486DX）から**数値演算コプロセッサ**と呼ばれる浮動小数点演算専用のLSIを省いたものになっています。また、80486の**内部クロック数**を2倍に速めたものを**80486DX2**と呼び、3倍に速めたものをIntelDX4（**80486DX4**）と呼んでいます。80486にも、節電機能をもたせた**80486SL**があります。

Pentium（**ペンティアム**）は、現在最も新しい高性能なCPUです。

同じCPUならクロック数が速いほうがよい

コンピュータは、**クロックパルス**と呼ばれる一定の信号によってタイミングをとりながら作動しています。そして、クロックパルスの速度（**クロック数**）が速まれば、それだけ処理速度が向上するようになっているのです。ですから、CPUを80386、80486、Pentiumというようにレベルアップするほかに、クロック数が上のものにするという方法でもよいことになります。

たとえば、CPUがクロック数33MHz（メガヘルツ）で動作するパソコンであれば、66MHzのCPUと交換すると、倍の処理速度が期待できることになります。ただし、2倍速タイプの80486DX2や、3倍速タイプのIntelDX4のように、内部のクロック数だけを速くしたものは、外部のクロック数は変わっていないため、30％程度のロスが発生します。

メモリとは

メモリとは、パソコン（コンピュータ）のデータやプログラムを記憶（保存）しておくための装置です。パソコン本体内にあるプログラムやデータの格納場所を特に**主記憶装置**と呼びます。

主記憶装置には、高速のICメモリが用いられています。なお、ICメモリのうち、あらかじめ必要なプログラムなどが書き込まれていて、読み出し専用でユーザーが書き換えることはできないものを**ROM**（**Read Only Memory**）、パソコンの電源をオンにしてから自由に読み書きできるものを**RAM**（**Random Access Memory**）といいます。

プログラムやデータは、フロッピーディスク、ハードディスク、RAMディスク、MO（光ディスク）などにも保存できますが、これらは**補助記憶装置**と呼ばれ、主記憶装置を補うものです。

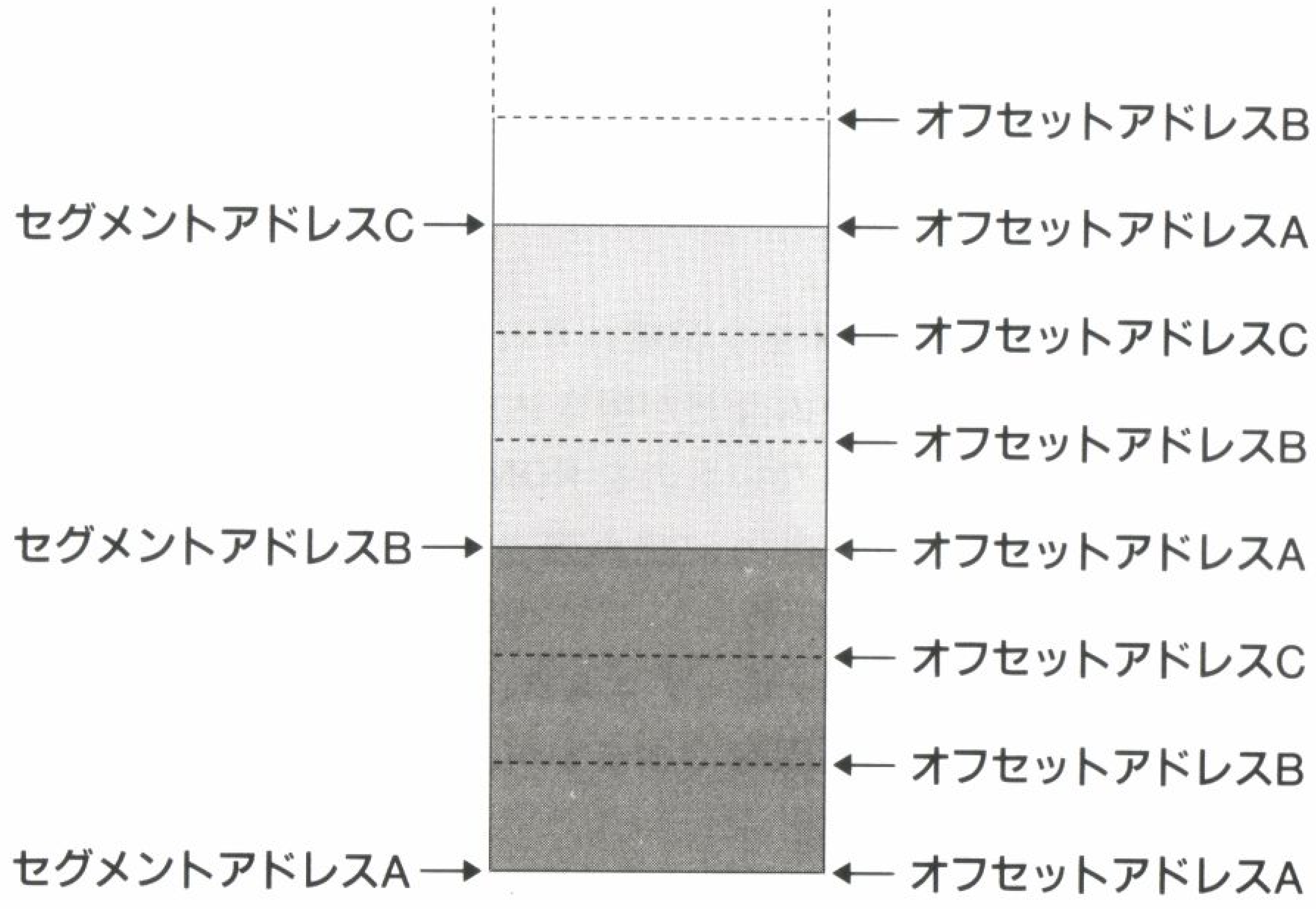
アドレスとは

CPUがメモリ内のデータに対して読み書きを行う場合、その場所を**アドレス**という番地で指定します。MS-DOSの対象となっている8086（互換CPU含む）というCPUには、アドレスを管理する信号線（**アドレスバス**）が20本あり、それぞれのアドレスバスで0と1の信号を扱えるため、全体で2の20乗（00000～FFFFFH：1Mバイト）のアドレスをもてることになります。このアドレスを**物理アドレス**といいます。

ところで、8086は16ビットCPUと呼ばれるとおり、一度に16ビットのデータ（0000～FFFFH）までしかやり取りできません。このため、プログラムやデータで直接物理アドレスを表現することはできないのです。そこで、1Mバイト（16進数の5桁）のアドレスを上位4桁（**セグメントアドレス**）と下位4桁（**オフセットアドレス**）に分け、この2つを加算して最終的な物理アドレスを求めるようになっています。セグメントアドレスとオフセットアドレスを使ったアドレス値を**論理アドレス**と呼び、このようなアドレスの管理方法を**論理アドレス方式**と呼びます。

また、基本（ベース）となるアドレス（セグメントアドレスのこと）からの位置（オフセット）で表すことから、**ベースアドレス方式**とも呼ばれます。

◉セグメントアドレスとオフセットアドレスの概念図



セグメントとオフセットによる物理アドレスの計算

たとえば、セグメントアドレス 1000H でオフセットアドレス 2D34H であるなら、物理アドレスは 12D34H になります。セグメントアドレスは、後ろに 0 を加えて計算する点がポイントです。

◉セグメント 1000H、オフセット 2D34H のときの物理アドレスの算出

セグメントアドレス		10000	(1 桁上げる)
オフセットアドレス	+)	2D34	
物理アドレス		12D34	

●１つのアドレスもいろいろに表現できる

同じアドレスでも、セグメントアドレスとオフセットアドレスの組み合わせによって、さまざまに表現することができます。

◉異なるセグメントとオフセットの組み合わせで同一アドレスを表現する

```
      11000        12D00        03000
+)     1D34   +)    0034   +)    FD34
   ---------    ---------    ---------
      12D34        12D34        12D34
```

ただし、通常はわかりやすいようにセグメントアドレスの最上位だけを変化させ、オフセットアドレスを0000～FFFFHの範囲で表現します。

物理アドレス 00000～0FFFFH	物理アドレス 10000～1FFFFH
セグメント：オフセット	セグメント：オフセット
0000：0000	1000：0000
⋮　⋮	⋮　⋮
0000：FFFF	1000：FFFF

Step 44

PC-9800 シリーズのメモリ構成

●ポイント●

・パソコンのメモリ領域は、コンベンショナルメモリ(基本メモリ)、アッパーメモリエリア、エクステンドメモリに分けられる。
・PC-9800 シリーズに MS-DOS を読み込んだ場合、メモリ領域には、MS-DOS システム部、COMMAND.COM 常駐部、ユーザーズメモリ、テキスト V-RAM、グラフィック V-RAM、拡張 ROM 空間、拡張グラフィック V-RAM、システム ROM、エクステンドメモリが割り当てられる。
・エクステンドメモリのうち、アドレス 100000～10FFEFh には、HMA メモリとして DOS のシステムの一部や、ディスクバッファ、DoubleSpace の一部を読み込むことができる。

●解 説●

CPU とメモリの関係

パソコンというのは、「プログラムがなければただの箱」と言われるくらい、マシン本体だけではほとんど何もできないしくみになっています。それは、パソコンが将来の拡張に備えるため、ほとんどのプログラムをオプションにしているからです。ワープロのように、最初からすべてのプログラムが本体に組み込まれてしまうと、プログラムの仕様が古くなったら、本体ごと買い替えなければなりませんが、パソコンなら、プログラムを交換するだけで済みます。

パソコンのプログラムのなかでも、**OS（Operating System）**と呼ばれるプログラムは、ワープロソフトや表計算ソフトのような、実用的なプログラムを使用するための基本的な環境作りを行うため、必ず最初に実行されます。OSの存在があるため、ほかのプログラムは専門の処理を行う機能を持つだけで済んでいるのです。

さて、われわれがPC-9800シリーズで使っている**MS-DOS**は、MicroSoft Disk Operating Systemの略で、Microsoft社がIBM社の依頼を受けて開発したOSです。MS-DOSは、もともと**8086**という16ビットCPU(CPUとはパソコンの頭脳にあたります）を搭載したマシン向けのOSであるため、**80486**や**Pentium**のような32ビットCPUが主流になっている現在では、制限ばかりが目立つOSになってしまいました。次期OSとされている**Windows95**が登場すれば、MS-DOSは使われなくなるか、または大きく仕様変更されることになるかもしれません。

MS-DOSが抱えている制限のなかで、現在、最も問題になっているのがプログラムを実行するメモリ領域の大きさです。

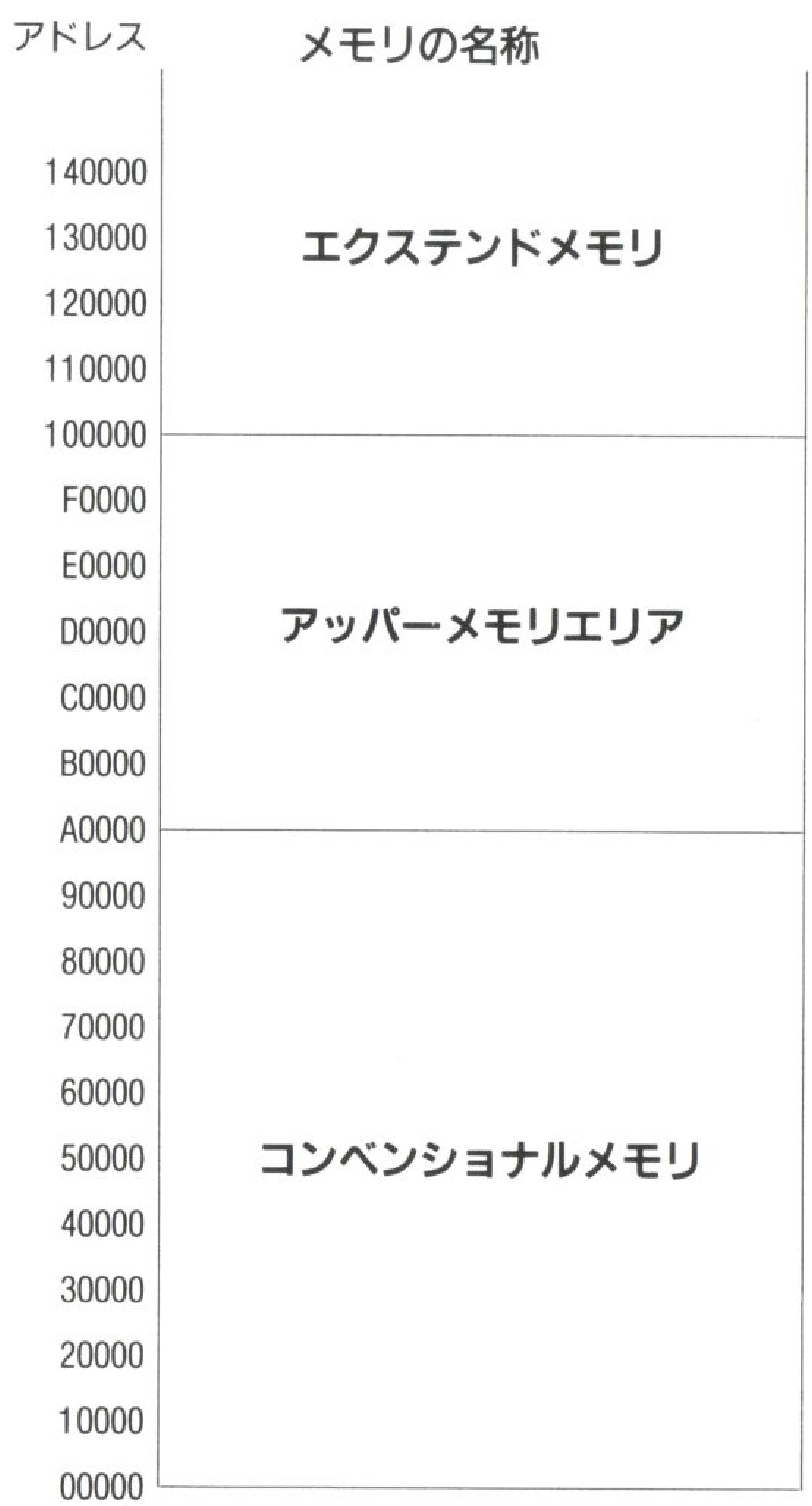
メモリには、**アドレス**と呼ばれる番地が割り当てられますが、CPUがアドレスを管理する線(**アドレスバス**)では、1本につき0か1という2種類の信号をやり取りします。8086は20本のアドレスバスを持っているので、2の20乗＝1Mバイト（メガバイト）のアドレスを管理できることになります。80486やPentiumには32本のアドレスバスがあるので(※)、本来なら2の32乗＝4GB（ギガバイト)のアドレスを管理できるのですが、MS-DOS環境で動作させる限り、通常の方法では8086並みのメモリしか利用できないのです。

※本書を執筆している時点での仕様です。

パソコンの基本的なメモリ構成

一般的なパソコンは、すべてメモリが次の3つの部分に分けられています。なお、この図のように、アドレスの利用状況を示したものを**メモリマップ**と呼んでいます。

◉ PC-9800シリーズのメモリマップ（ノーマルモード）



●コンベンショナルメモリ

コンベンショナルメモリは、**メインメモリ**、**基本メモリ**とも呼ばれ、ユーザーがプログラムを読み込んで自由に使うことのできる領域です。PC-9800シリーズでは、ノーマルモードで640Kバイト、ハイレゾリューションモードで768Kバイトあります。ただし、MS-DOSのシステムもここに読み込んで使用するため、実際にユーザーが使えるメモリは、もっと少なくなります。

●アッパーメモリエリア

コンベンショナルメモリの続きから、1Mバイトまでの領域が**アッパーメモリエリア**（**上位メモリ領域**）になります。本体の背面にある拡張スロットにボード類を差し込んで使用する場合、そのボード上のROM(読み出し専用のメモリ）を読み出すために、この領域が割り当てられます。

PC-9800シリーズでは、本来はユーザーが使用できないアッパーメモリエリアを搭載メモリに含めないため、3.6Mバイトや7.6Mバイトというように、コンベンショナルメモリと拡張メモリを足した分が搭載メモリとされています。PC/AT互換機(DOS/V機)では、アッパーメモリエリアも含めたメモリを搭載メモリとするため、4Mバイト、8Mバイトとなっています。

EMSという拡張メモリのための仕様では、この領域のうちの未使用の部分にページフレーム(物理ページ)を作成します。また、XMSという拡張メモリ仕様では、未使用の部分をUMBメモリとして割り当て、ユーザーがデバイスドライバや常駐プログラムを読み込むことができるようにします。

この領域の使い方次第で、コンベンショナルメモリの自由な空間を広げることができます。

●エクステンドメモリ

80286以降のCPUは1Mバイト以上のアドレスを管理できるのですが、それまでの8086と互換性をもたせるため、1Mバイトを超えるメモリは**エクステンドメモリ**（**拡張メモリ**）として特別扱いすることになっています。

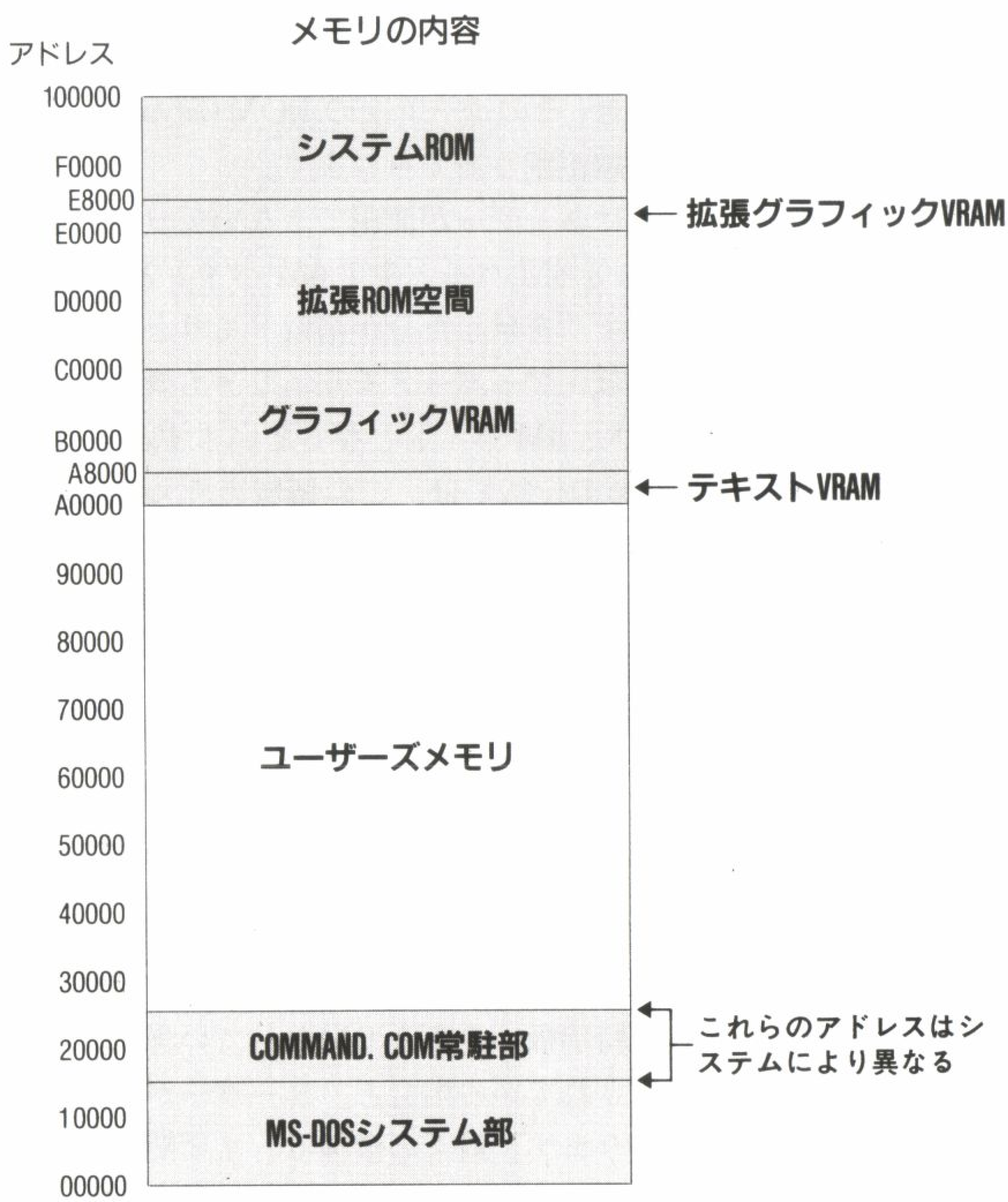
拡張メモリは、最初から搭載されている場合と、ユーザーがメモリを買い足さなければならない場合とがあります。最近の機種は、まず間違いなく拡張メモリが搭載されていますが、Windows環境で利用するつもりなら、10Mバイト以上の拡張メモリになるようにしておくことをお勧めします。

拡張メモリを使用するには、**拡張メモリマネージャ**と呼ばれる管理プログラムが必要になります。

PC-9800 シリーズのメモリ構成

PC-9800 シリーズに MS-DOS のシステムを読み込んで使用する場合、コンベンショナル、アッパーメモリエリア、エクステンドメモリは次のように利用されます。

◉ PC-9800 シリーズのメモリマップ（詳細）



MS-DOS システム

この部分をさらに詳しく見ると、4 つの部分に分けることができます。

◉ MS-DOS システム部



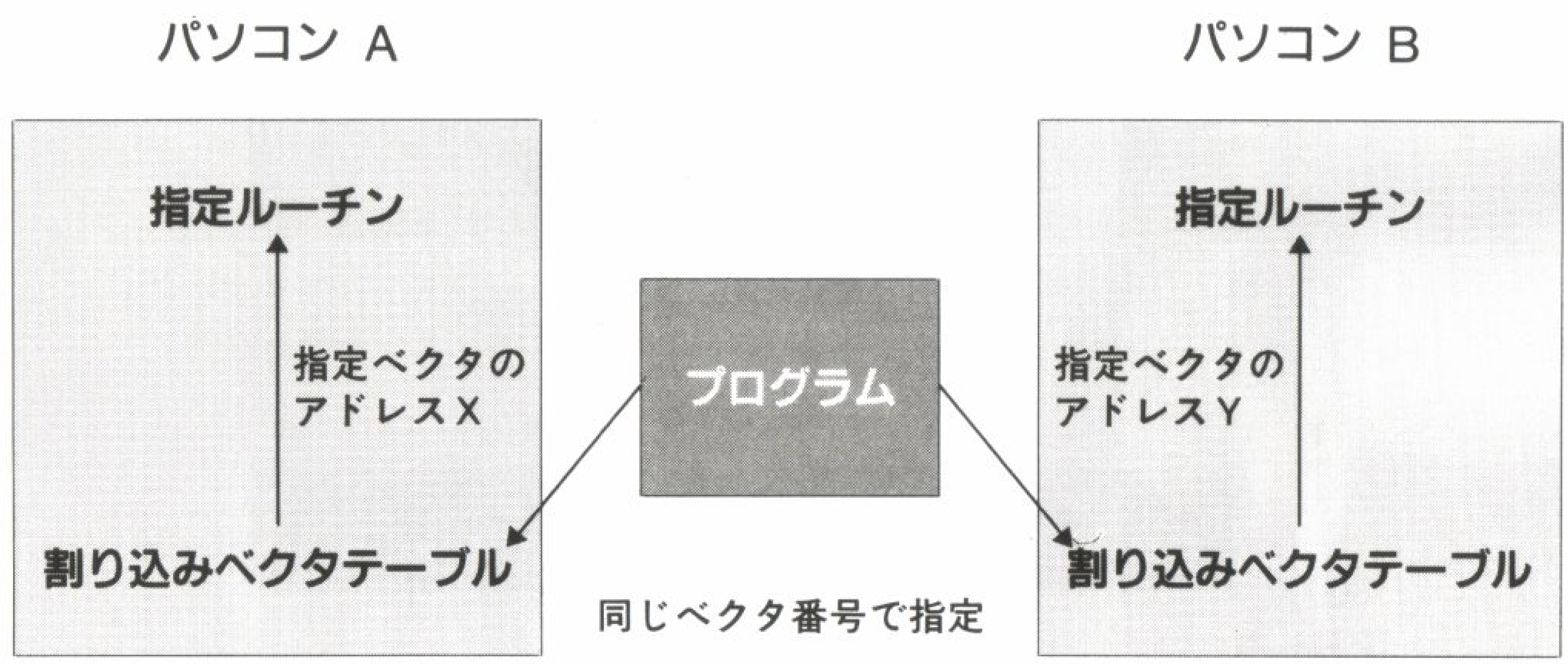
●割り込みベクタテーブル

1 つの処理を実行している最中に割り込んで、別の処理を行うことを**割り込み**と呼びます。個々の割り込み手順(ルーチン)は ROM 部分や MS-DOS システム内部、デバイスドライバ内部などに置かれ、それぞれに「**ベクタ**」と呼ばれるコード番号が割り当てられています。

この**割り込みベクタテーブル**には、コード番号に対応したルーチンの開始アドレスが記録されていて、プログラムがコード番号を指定すると、希望するルーチンが実行されるようになっています。

MS-DOSが持っているこの仕組みのおかげで、異なる機種でも同じプログラムを動作させることができるのです。

◉割り込みベクタテーブルの仕組み



● IO.SYS

MS-DOSを起動するために必要な**システムファイル**のひとつである**IO.SYS**からは、基本デバイスドライバなど、各種ハードウェアを直接操作するためのプログラムが読み込まれています。

IO.SYSはハードウェアを直接操作する原始的な部分で、**MSDOS.SYS**によって呼び出されます。

● MSDOS.SYS

ファイル、メモリ、各種周辺機器を操作するプログラムで、外部プログラムとIO.SYSの仲介役も果たします。

● CONFIG.SYS設定部

CONFIG.SYS内で設定したFILES、FCBS、BUFFERS、LASTDRIVEの必要メモリを確保します。特に指定がなければ、それぞれの初期値に従って設定されます。

また、CONFIG.SYSのDEVICEコマンドで組み込んだデバイスドライバも、この部分に常駐しています。

この部分だけは、ユーザーがCONFIG.SYSの設定を変えることによって、占有メモリの大きさを変更することが可能です。

COMMAND.COM常駐部

CONFIG.SYS内のSHELLコマンドによってシステムに常駐している**シェルプログラム**です。シェル（shell）には、貝、外郭という意味がありますが、シェルプログラムは、コンピュータのシステムを包み込んで、人間が使いやすいように手助けする役目を果たしています。

MS-DOSの**内部コマンド**であるDIRコマンドやCOPYコマンドは、このCOMMAND.COMが持っている機能です。

ユーザーズメモリ

フリーメモリ、**空きメモリ**とも呼ばれる部分で、ユーザーが実行したプログラムはここに読み込まれてから処理されます。この部分の容量によって、起動できるプログラムの数や大きさが決まってきます。

テキストVRAM（T-VRAM）

VRAMというのは**ビデオRAM**のことで、画面に表示するデータを書き込むためのメモリです。

PC-9800シリーズの画面表示機能には、主に文字だけを表示するテキストモードと描画が可能なグラフィックモードとがあります。そのうち、テキストモードでの表示をするためのデータを書き込むメモリを**テキストVRAM**（**T-VRAM**）といいます。

この領域のなかには、UMBメモリとして割り当てても問題が起こらない部分があります。

グラフィックVRAM（G-VRAM）

グラフィック表示のデータを書き込むメモリです。ごく一部の旧機種を除き、PC-9800シリーズでは同一アドレスに2組の**VRAM**が装備されていて、切り替えて表示することができるようになっています。

また、T-VRAMと**G-VRAM**（**グラフィックVRAM**）が別にあるため、テキスト画面とグラフィック画面を合成して表示することもできるのです。

拡張 ROM 空間

本体の背面にある**拡張用スロット**に差し込んで利用する RAM ボード、FM 音源ボード、SCSI ボードなどの各種拡張ボードにある ROM が、この部分のアドレスを通して利用されます。

アドレス C0000～C8000H までを、特に**ユーザー用拡張 ROM 空間**と呼びます。

この領域のうち、未使用の部分を UMB メモリとして設定すると、ユーザーズメモリを補うためのメモリとして使用することができます。

拡張グラフィック VRAM（拡張 VRAM）

通常、グラフィック画面は 8 色で表示されますが、16 色表示まで拡張されるとき、この VRAM が利用されます。

システム ROM

N88-BASIC システムが置かれている **ROM**（読み出し専用メモリ）部分です。ここに含まれているルーチン（プログラム）にベクタコードを割り当て、ソフトウェア割り込みによって呼び出すことができます。

この領域のなかには、UMB メモリとして割り当てても問題が起こらない部分があります。

エクステンドメモリと HMA

1M バイトを超えるメモリは、**エクステンドメモリ**（**拡張メモリ**）と呼ばれ、EMS、XMS、VCPI、DPMI などの特別な手法によって利用することができます。

エクステンドメモリのうち、アドレス 100000～10FFEFh の範囲を、特に **HMA**（**High Memory Area**）メモリと呼びます。HMA メモリには、DOS6 のシステムの一部や、ディスクバッファ、DoubleSpace の一部を読み込むことができます。ただし、HMA メモリを利用できるのは 1 つのプログラムに限られるため、それらのメモリを割り当てた場合には、ほかのプログラム (FEP など) で利用することはできません。

Step 45

拡張メモリ仕様

●ポイント●

- 主な拡張メモリ仕様には、EMS、XMS(HMA、UMB)、VCPI、DPMI がある。
- DOS6 は EMS、XMS(HMA、UMB)、VCPI に対応している。

●解 説●

640K バイトの壁を打ち破る拡張メモリ仕様

すでに解説したように、MS-DOS は 16 ビットパソコン用の OS であるため、1M バイトまでのメモリにしか対応していません。しかも、そのうちの 384K バイト（ノーマルモードの場合）をシステムが必要とするため、本当に利用できるのは 640K バイトとなっています。

この「640K バイトの壁」は、大規模なソフトウェアを開発する上で大きな足かせとなってきましたが、近年 80286 以上の CPU を持った機種が主流となり、その機能を利用してメモリを拡張するための手法も考えられました。それが EMS、XMS(HMA、UMB を含む)、VCPI、DPMI などの**拡張メモリ仕様**です。

ここで、それぞれの仕様について簡単に説明しておきましょう。

EMSとは

EMS（**Expanded Memory Specification**：拡張メモリ仕様）は、アメリカのロータス社とインテル社がIBM-PC/ATのPC-DOSを対象にして1985年に発表したものです。

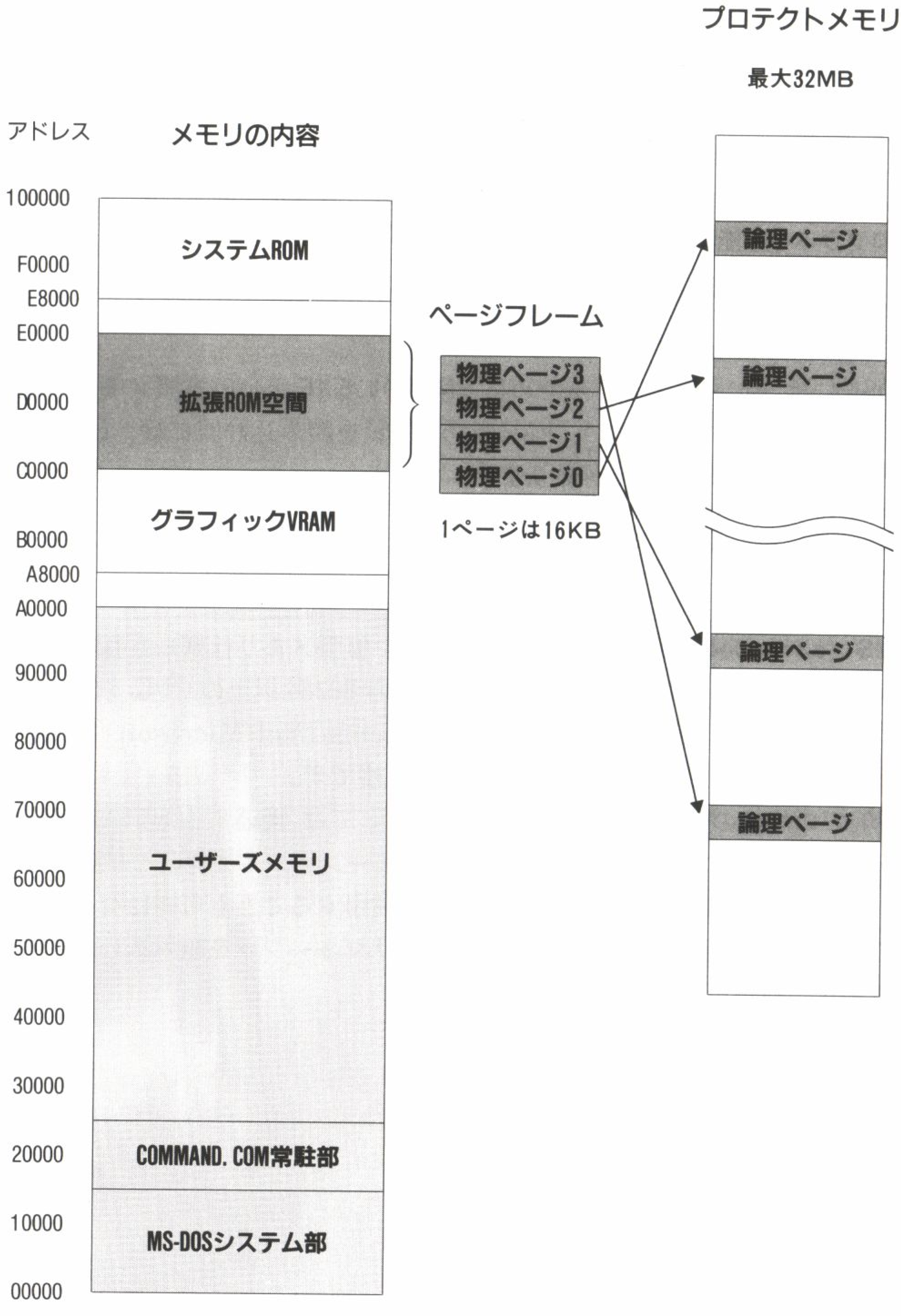
その後、マイクロソフト社が加わって、**LIM EMS3.2**という規格になりました（LIM＝Lotus/Intel/Microsoft)。最新バージョンは4.0で、**LIM EMS4.0**と呼ばれています。

● EMSのしくみ

EMSでは、16Kバイトの**ページ**という単位でメモリを管理しています。そして、数ページからなる**ページフレーム**と呼ばれる窓を通して、対応するアプリケーションがデータやプログラムを参照するのです。ページフレームで使われるページを、特に**物理ページ**といいます。

一方、切り換え用に使う拡張されたメモリは**論理ページ**で分割されて、必要に応じて物理ページと入れ替えられるようになっています（この入れ替え動作を**マッピング**と呼びます)。

EMSは、専用のボードを使えばCPUに8086が使われた機種でも利用できたため、最も普及が早かった拡張メモリ仕様です。現在では、80386以上のCPUがもっている**仮想8086モード**という特殊な機能を利用した、**仮想EMS**という手法が使われているため、8086や80286を搭載したシステムに比べてたいへん高速になっています。


プロテクトメモリ
最大32MB
アドレス
メモリの内容
100000
F0000
E8000
E0000
D0000
C0000
B0000
A8000
A0000
90000
80000
70000
60000
50000
40000
30000
20000
10000
00000
システムROM
拡張ROM空間
グラフィックVRAM
ユーザーズメモリ
COMMAND.COM常駐部
MS-DOSシステム部
ページフレーム
物理ページ3
物理ページ2
物理ページ1
物理ページ0
1ページは16KB
論理ページ
論理ページ
論理ページ
論理ページ

●3.2と4.0の違い

LIM EMS3.2とLIM EMS4.0の違いは、主に次の点にあります。

- **使用可能な拡張メモリが3.2では8Mバイト、4.0では32Mバイトまで。**
- **窓口となるページフレームが3.2では連続した4ページと決められていたが、4.0では8ページ以内で自由となり、連続していなくても許される。**
- **EMSメモリとして利用する内容が、3.2ではデータ中心であったが、4.0ではプログラムも可能になっている。**

現在のところ、EMSドライバはLIM EMS4.0に対応していても、実際には3.2の範囲内で利用されています。

DOS6には、EMSを実現するために**EMM386.EXE**という**拡張メモリマネージャ**が付属になっていますが、EMM386.EXEを組み込む前には、HIMEM.SYSを組み込んでおく必要があります。

XMSとは

XMS（**eXtended Memory Specification**：拡張メモリ仕様）も基本的にはEMSと同様の拡張メモリ仕様ですが、最初から80286以上のCPUを持った機種を対象にしています。EMSの3社（Lotus/Intel/Microsoft）にAST Research社を加えた4社が提携して作った規格です。

当初、この規格のうちで使われていた機能としては、後述するHMAとUMBに限られていましたが、DOS5からはアプリケーションさえ対応していればデータやプログラムなどを拡張メモリに置いて利用することも可能になりました。

DOS6では、**HIMEM.SYS**という拡張メモリマネージャを組み込むことにより、XMSが利用可能になります。

HMAとは

XMSでは、エクステンドメモリのうち物理アドレス100000～10FFEFHまでを**HMA**（**High Memory Area**）、それ以上のアドレスにあるメモリを**EMB**（**Extended Memory Block**）と呼んでいます。

どうしてHMAがそれ以上の領域と区別されているか説明しましょう。

8086には、アドレスを管理するアドレスバスが20本あり、それぞれのアドレスバスで0と1の信号を扱えるため、全体で2の20乗(00000～FFFFFH:1Mバイト）のアドレスを持てることになります。このアドレスを**物理アドレス**と呼びます。

ところで、前述したとおり、8086は16ビットのデータ（0000～FFFFh）しか扱えませんから、プログラムやデータで直接物理アドレスを表現することはできません。そこで、1Mバイト（16進数の5桁）のアドレスを上位4桁のセグメントアドレスと下位4桁のオフセットアドレスに分け、この2つを加算して最終的な物理アドレスを求めています。

さて、セグメントアドレスをFFFFhとした場合、オフセットアドレスを0010Hに設定すると、物理アドレスが100000hとなり、8086が持つ20本のアドレスバスでは管理できなくなってしまいます。

```
    FFFF0             1111 1111 1111 1111 0000
+)   0010→      +)         0000 0000 0001 0000
   100000           1 0000 0000 0000 0000 0000
                    ↓
            このビットが無効になる
```

このため、このようなアドレス指定をすると、8086では物理アドレスは0に戻ってしまいます。しかし80286以上のCPUではアドレスバスが24本以上あるため、一時的に21本目のアドレス線（番号が0番から付けられるため20番となる）を有効にしてやることで、このアドレス指定を使えるようになります。つまり、本来はエクステンドメモリに属する、セグメントアドレスFFFFh、オフセットアドレス0010～FFFFhの領域（物理アドレス100000～10FFEFh）を、通常のモードで利用できることになるのです。

そこでXMSでは、この65520バイト（約64Kバイト）のメモリをHMAと呼び、使用許可が与えられた1つのプログラムに対してのみ、アドレスバス20番をオンにして利用可能な状態にしています。

HMAはXMSの一部分にすぎないのですが、当初はHMAを利用することをXMSと呼んでいました。しかし、DOS5からはXMSを本来の広い意味で使うようになっているため、HMAと区別して呼ぶようになりました。

DOS6では、HMAにシステムの一部を入れて利用することができますが、その場合にはCONFIG.SYSで、

```
DOS=HIGH
```

と記述します。

UMBとは

UMB（Upper Memory Block）は、**アッパーメモリエリア**（物理アドレスA0000～FFFFFh）のうち、ユーザーが使用しても問題がない部分にエクステンドメモリを割り当てて、デバイスドライバや常駐プログラムを組み込むために利用できるという機能です。ただし、アッパーメモリエリアはEMSのページフレームにも使いますから、EMSと同時に使うことになると、UMBメモリとして使える領域が64Kバイト減ることになります。

現在のシステムでは、UMBメモリの活用がコンベンショナルメモリの節約に最も効果があると言えるでしょう。

DOS6でUMBメモリを有効にするためには、CONFIG.SYSでEMM386.EXEに/UMBスイッチを付加し、さらに、

```
DOS = UMB
```

と記述します。

通常はHMAと同時に利用することが多いので、その場合は

```
DOS = HIGH,UMB
```

のように指定します。

VCPIとは

DOS環境で1Mバイトを超えるメモリを使用するため、まず、**DOSエクステンダ**（Phar Lap Software,Inc.他）というプログラムが開発されました。しかし、仮想EMSという手法が主流となり、似たような機能をもったDOSエクステンダと衝突するようになってしまったのです。

そこで、DOSエクステンダを開発したPhar Lap Software,Inc.とQuarterdeck Office Systemsが共同で、DOSエクステンダと仮想EMSを一緒に管理できる**VCPI（Virtual Control Program Interface）**という規格を作りました。VCPIに対応したアプリケーションは非常に少ないのですが、たとえば、『一太郎 Ver.5』などは、VCPIと後述するDPMIに対応しています。

DOS6に付属のEMM386.EXEは、/DPMIスイッチを付けるとVCPIにも対応します。

DPMIとは

VCPIによって仮想EMSとDOSエクステンダは共存できるようになりましたが、VCPIは**シングルタスク**を前提とした規格であるため、386拡張モードにおけるWindowsのように、複数のプログラムが同時に実行される**マルチタスク**の環境には対応できません。

そこで、Intel Corporationが中心となって、新たに**DPMI（DOS Protected Mode Interface）**という規格が作成されたのです。Windowsのバージョン3.0からは、このDPMIが採用されています。

DPMIでは、DOSエクステンダを含むDPMI対応アプリケーションを**DPMIクライアント**、DPMI環境を提供するプログラムを**DPMIサーバ**と呼びます。

DOS環境でDPMI対応ソフトを利用するためには、EMM386.EXEに/DPMIスイッチを付けて、さらにDPMI.EXEを組み込む（または実行する）必要があります。

Step 46

割り込みレベル（IRQ）、I/O ポート、DMA

●ポイント●

・割り込みレベルとは、割り込みコントローラが判断する割り込み処理の優先順位のこと。
・IRQ とは、各 LSI から CPU に出される割り込み要求信号のこと。
・割り込みレベルは 0〜15 まであり、小さいほど優先順位が高い。
・割り込みレベル（IRQ）のうち、7 番と 15 番はシステム予約となっていて使用できない。
・I/O ポートとは、各 LSI をコントロールするための出入り口。
・各 LSI 用の I/O ポートには、I/O ポートアドレスが割り当てられている。
・DMA とは、CPU を介さずにデータ転送を行うこと。
・DMA チャネルには 0〜3 番の 4 種類がある。
・割り込みレベル（IRQ）、I/O ポートアドレス、DMA チャネルの設定は、使用している拡張ボードや PC カードのそれぞれで重複しないように注意する。

●解 説●

割り込みレベルとは

パソコンの頭脳にあたるCPUでは、メモリからプログラムを読み込んで、その内容を処理しています。しかし、プログラムを実行している最中に、もっと優先順位が高い処理要求があると、プログラムの実行を中断して、要求のあった処理を行うようになっています。これを、**割り込み**または**割り込み処理**と呼んでいます。これにより、CPUが効率よく処理できるほか、複数の処理をほぼ同時に実行することができるのです。

さて、パソコンの内部にはハードディスクやフロッピーディスクのコントロールなど、さまざまな役割をもったLSI（大規模集積回路）が組み込まれていて、各LSIからプログラム実行中のCPUに対して、**割り込み要求信号**（**Interrupt request signal：IRQ**）が送られてきます。いくつもの信号が送られてきても、CPUが混乱することなく処理を行えるのは、信号の優先順位を判定して、優先順位の高い信号から順に送り出してくれる**割り込みコントローラ**という補佐役のLSIが付いているからです。

PC-9800シリーズは、8つのレベルを判定できる割り込みコントローラを2個備えていて、16種類のIRQに対して0〜15までのレベル（＝**割り込みレベル**）を設定できます。レベルは、小さいほど優先順位が高くなります。ただし、レベル7とレベル15はシステム予約となっているため、実際に有効なのは14レベルとなっています。

次の表は、割り込みレベル(IRQ)と、それに割り当てられている割り込み名の一覧です。

●割り込みレベル（IRQ）一覧

レベル	IRQ	割り込み名
0	0	タイマ
1	1	キーボード
2	2	CRTV
3	3	拡張バス　INT 0
4	4	RS-232C
5	5	拡張バス　INT 1
6	6	拡張バス　INT 2
7	7	（スレーブ）
8	8	プリンタ／NDP
9	9	拡張バス　INT 3（ハードディスク）
10	10	拡張バス　INT 41（640Kバイトフロッピーディスク）
11	11	拡張バス　INT 42（1Mバイトフロッピーディスク）
12	12	拡張バス　INT 5
13	13	拡張バス　INT 6（マウス）
14	14	NDP
15	15	（システムタイマ）

※ NDP：数値演算コプロセッサ

多くの割り込みレベルがシステムで使われているため、拡張ボードなどに割り当て可能なのは、レベル3、5、6、12くらいなものです。最近では、640Kバイトフロッピーディスクは使われなくなってきているため、レベル10もよく利用されています。

拡張ボードやPCカードのマニュアルで、「割り込みレベル」または「IRQ」とあったら、この表を思い出してください。割り込みレベルやIRQのほか、対応する「INT～」の部分で指定するようになっているものもあります。選択可能な範囲から、ほかの拡張ボードなどで使用していないものを選びますが、初期設定のままでは重複してしまうケースも多いので、慎重に選ぶようにしてください。

I/O ポートとは

パソコンにはさまざまなLSIが使われていますが、それらをコントロールしやすいように系統的にまとめて、制御信号の出入り口を決めたものが**I/O ポート**です。I/O とは input と output、つまり入出力のこと。ポート(port)には港という意味があります。

CPU は、I/O ポートに割り当てられたアドレス（=**I/O ポートアドレス**）に対して制御データを送り、各 LSI をコントロールします。また、ユーザーもプログラムのなかでI/O ポートアドレスに制御データを送るようにすれば、LSI のコントロールが可能になるのです。

拡張ボードや PC カードで、「I/O ポートアドレス」に関する設定があったら、選択可能な範囲のなかから、それまでにセットした拡張ボードなどに割り当てていないアドレスを指定するようにしてください。

DMA とは

通常、周辺装置（LSI）とメモリの間でデータを転送する場合には、CPU が制御することになっています。しかし、大量のデータを転送すると、CPU にかかる負担が大きくなって、システムの効率が低下することになってしまいます。

そこで、CPU を介さないデータ転送（direct memory access：**DMA**）ができるように、パソコンにはデータ転送専用の LSI が搭載されています。それが、**DMA コントローラ**です。

PC-9800 シリーズの DMA コントローラは、4 つの独立した通路(=**DMA チャネル**)を使ってデータ転送を行います。それらには 0〜3 番までの番号が付けられていて、0 番の優先順位が最も高くなっています。

拡張ボードや PC カードで、「DMA」または「DMA チャネル」に関する設定があったら、選択可能な範囲のなかから、それまでにセットした拡張ボードなどに割り当てていない番号を指定するようにしてください。

DMA を用いるボード類はそれほどありませんが、なにしろ用意されている DMA チャネルが 4 種類しかなく、そのうち拡張スロットなどのためにユーザーに解放されているのは 0 番と 3 番のみになっています。くれぐれも重複させることのないように注意してください。

Step 47

MemMakerによるメモリの最適化

●ポイント●

- MemMakerはMEMMAKERコマンドで起動する。
- MemMakerによる最適化は、カスタムセットアップで行うほうがよい。
- MemMakerによる最適化がうまくゆかなければ、/UNDOスイッチを付けて実行すると、直前のCONFIG.SYS／AUTOEXEC.BATに戻る。

●解 説●

MemMakerによる最適化

DOS5からは、ユーザーが積極的に拡張メモリを利用するためのコマンドがいくつかサポートされました。しかし、そのためにCONFIG.SYSやAUTOEXEC.BATの設定が複雑化してしまい、初心者には難しいOSになってしまった感も否めません。

その点、DOS6には**MemMaker**というユーティリティが付属になったため、これまで初心者には不可能だったレベルでのCONFIG.SYSやAUTOEXEC.BATの最適化が可能になったのです。自分で使うCONFIG.SYSやAUTOEXEC.BATができあがったところで、このMemMakerコマンドを実行するとよいでしょう。

MemMakerは、**MEMMAKER**コマンドによって設定を行います。

なお、最適化を行いたいCONFIG.SYSとAUTOEXEC.BATには、それぞれDEVICEHIGHコマンド、INSTALLHIGHコマンドや、LOADHIGH(LH)コマンドを用いていても、いなくてもかまいません。

ただし、AUTOEXEC.BATの最後にアプリケーションを起動するようになっていたら、その部分はコメントにしておいてください。

MemMakerの操作手順

次のように入力するとMemMakerが起動します。

```
A:¥>MEMMAKER
```

以下、次の手順で実行します。

①セットアップ方法の選択

オープニング画面で⏎キーを押すと、**高速セットアップ**で実行するかどうかの問い合わせがあります。ここでは、スペースキーを押して「**カスタムセットアップ**」に変更してから、⏎キーを押してください。高速セットアップを選択すると、標準的な設定で実行されますが、それではほとんど最適な環境は得られません。

② EMSメモリの使用

まずEMSメモリを利用するかという問い合わせがあります。DOS環境では、必ず「はい」を指定しておきましょう。Windows環境のみで利用するなら、「いいえ」にしておくとUMBメモリが多く確保できます。これは、EMM386.EXEに/UMBスイッチを付けるか、/NOEMSスイッチを付けるかという設定になります。

③最適化条件の指定

最適化に関して、以下の条件を設定します。
「いいえ」と「はい」は、スペースキーで変更できます。設定が完了したら、⏎キーを押してください。

・最適化に含めるドライバとTSRを指定しますか?

TSRとは常駐プログラムのことです。UMBメモリに移動してはならないプログラムが含まれている場合には、「はい」を指定して、いちいち指定するほうがよいでしょう。ただし、MEMMAKER.INFでUMBメモリへの組み込みが禁止されているプログラムに関しては、最初から選択に含まれないようになっています。

最初は、「いいえ」の状態で実行してください。

・上位メモリ領域を積極的にスキャンしますか?

アッパーメモリエリア(上位メモリ)にあるアドレスE8000-F3FFFhの範囲のうち、未使用の領域をUMBに割り当てるかどうかの指定です。これは、EMM386.EXEに/HIGHSCANスイッチを付けるかどうかの指定にあたります。

特に不具合が生じない限り、ここは「はい」で実行してください。

・上位メモリ領域をWindowsで使うために最適化しますか?

Windows環境からDOSアプリケーションを実行する際に用いる変換バッファのために、アッパーメモリエリアから2つの領域を予約しておくかどうかの指定です。この指定によって、EMM386.EXEの/WINスイッチによるメモリの確保が行われます。

/WINスイッチに指定されたアドレスは、DOS環境でUMBメモリとして利用することができなくなります。Windows環境で主にDOSのアプリケーションを起動する場合にのみ、「はい」を選択するようにしてください。DOS環境だけで利用する場合、あるいは、Windows環境で主にWindows対応のアプリケーションだけを利用する場合には、「いいえ」を指定しておきましょう。

・現在のEMM386メモリ設定をこのまま保持しますか?

現在のCONFIG.SYSで、EMM386.EXEに付けられているオプションをそのまま引き継ぐかどうかの指定です。/UMBスイッチや/NOEMSスイッチ以外に、/Eスイッチなどを指定してある場合には、「はい」にしておくほうがよいでしょう。

設定をMemMakerに任せるなら「いいえ」にしておきます。

④ Windowsディレクトリの指定

各ドライブが検索され、Windowsがインストールされていると判断されたディレクトリがあれば、それが表示されます。

正しいディレクトリであれば、そのまま⏎キーを押してください。

誤っていれば、変更してから⏎キーを押します。

⑤再起動とCONFIG.SYS／AUTOEXEC.BATの設定

再起動をするという表示に⏎キーを押すと、システムがリセットされ、自動的にMemMakerが起動されます。

ここでは、最適な設定が検討され、⏎キーを押してゆくと、再びシステムがリセットされます。この再起動の際には、画面に表示されるメッセージに注意してください。何らかのエラーメッセージがあれば、設定に問題があると考えられます。

再び、MemMakerが起動され、何か問題があったかどうかという問い合わせをしてきます。問題がなければ、そのまま⏎キーを押してください。

メッセージ画面のあと、実行前後のメモリの比較結果が表示されます。「使用後」の部分にある「空きコンベンショナル（基本メモリ）」が大きな値になっていれば、最適化の効果があったということになります。

⏎キーを押すと、MemMakerは終了します。

MemMakerによる最適化が失敗した場合の対処方法

MemMakerが正常に終了しなかった場合には、おそらくUMBメモリに読み込んではいけないドライバがあったと考えられます。再びMemMakerを起動して、「最適化に含めるドライバとTSRを指定しますか?」の部分であやしいと思われるドライバまたは常駐プログラムを最適化から除外するように設定してください。特に、日本語FEPは引っ掛かっている可能性が大です。

なお、CONFIG.SYSとAUTOEXEC.BATの内容をMemMakerの実行前に戻すには、/UNDOスイッチを付けて実行してください。

◉ MemMakerの設定前の環境に戻す

```
A:¥>MEMMAKER /UNDO
```

Step 48

MemMakerによる最適化の結果

●ポイント●

- MemMaker実行後のCONFIG.SYSとAUTOEXEC.BATは、DEVICEHIGHコマンド、LOADHIGH(LH)コマンドにUMBメモリの組み込みブロックと必要メモリのサイズが書き加えられるなどの変更が加えられている。
- MemMakerによって指定されたUMBメモリに関する値は、勝手に変更してはいけない。
- MemMakerによって設定されたFCBSの値は、手直しするほうがよい。

●解 説●

MemMaker 実行後の CONFIG.SYS

MemMaker を実行すると、CONFIG.SYS と AUTOEXEC.BAT が自動的に変更されています。

次の CONFIG.SYS を用いて最適化を実行してみます。

◉ MemMaker 実行前の CONFIG.SYS

```
BUFFERS=1
FILES=40
DOS=HIGH,UMB
LASTDRIVE=Z
SHELL=A:¥COMMAND.COM A:¥ /P
DEVICE=A:¥DOS¥HIMEM.SYS /TESTMEM:OFF
DEVICE=A:¥DOS¥EMM386.EXE /UMB /E=DC00-DFFF
DEVICEHIGH=A:¥DOS¥SETVER.EXE
DEVICEHIGH=A:¥DOS¥SSDRV.SYS
DEVICEHIGH=A:¥DOS¥CS.EXE
DEVICEHIGH=A:¥DOS¥CSALLOC.EXE A:¥DOS¥CSALLOC.INI
DEVICEHIGH=A:¥DOS¥CDCOMP.SYS
DEVICEHIGH=A:¥DOS¥MCDRV.SYS
DEVICEHIGH=A:¥DOS¥HDCARD.SYS
DEVICEHIGH=A:¥DOS¥CARDID.EXE A:¥DOS¥CARDID.INI
DEVICE=A:¥DOS¥RAMDISK.SYS 3072
```

結果は、次のようになりました。

◉ MemMaker 実行後の CONFIG.SYS

```
DEVICE=A:¥DOS¥HIMEM.SYS /TESTMEM:OFF
DEVICE=A:¥DOS¥EMM386.EXE /UMB /HIGHSCAN /E=DC00-DFFF
BUFFERS=1,0
FILES=40
DOS=UMB
LASTDRIVE=Z
FCBS=16,8
DOS=HIGH
SHELL=A:¥COMMAND.COM A:¥ /P
DEVICEHIGH /L:1,12560 =A:¥DOS¥SETVER.EXE
DEVICEHIGH /L:1,6496 =A:¥DOS¥SSDRV.SYS
DEVICE=A:¥DOS¥CS.EXE
DEVICEHIGH /L:1,23568 =A:¥DOS¥CSALLOC.EXE A:¥DOS¥CSALLOC.INI
DEVICEHIGH /L:1,9264 =A:¥DOS¥CDCOMP.SYS
DEVICEHIGH /L:1,11904 =A:¥DOS¥MCDRV.SYS
DEVICEHIGH /L:1,8688 =A:¥DOS¥HDCARD.SYS
DEVICEHIGH /L:2,31776 =A:¥DOS¥CARDID.EXE A:¥DOS¥CARDID.INI
DEVICEHIGH /L:2,19824 =A:¥DOS¥RAMDISK.SYS 3072
```

変更された箇所を検討していきましょう。

● HIMEM.SYS／EMM386.EXE

HIMEM.SYS と EMM386.EXE は、間違いなくほかのドライバよりも先に組み込まれるように、先頭に移動されています。

「上位メモリ領域を積極的にスキャンしますか?」という問いに「はい」としておいたので、EMM386.EXE には/HIGHSCAN スイッチが付けられています。これにより、UMB メモリが 44〜48K バイト増えているはずです。

この例では、PC カード用のドライバを組み込むため、あらかじめ/E スイッチでメモリを確保してあります。

● BUFFERS

BUFFERSの値のあとに、2次バッファを0にする指定が加えられています。ただ、2次バッファの既定値は0なので、これは指定しなくても同じことです。まぎらわしければ、通常の指定に戻しておきましょう。

● DOS

DOSコマンドは、「DOS=HIGH,UMB」としてあったものが、2つに分けられています。どちらにしておいても同じことですが、これも元の記述に戻しておくほうがわかりやすいかもしれません。

● FCBS

MemMakerを実行するとFCBSの設定が加えられますが、これははっきり言って余計なお世話です。この例のように、既定値の4よりも大きな値で設定されることもありますから、その場合にはFCBSの記述を削除するか、または最低の値になるように、「FCBS=1」と改めておきましょう。

なお、カンマで区切られた2番目の値は、FCBを用いてオープンしたファイルのうちで、クローズしてはいけないファイル数の指定です。マニュアルにも、HELPコマンドにも説明はありませんが、どういうわけかMemMakerを実行すると指定されるようになっています。省いてしまっても、まったく問題はありません。

● DEVICEHIGH /L:?,????=～

MemMakerによる最適化の結果、デバイスドライバがなるべく多くUMBメモリに収まるようにするための指定が、DEVICEHIGHコマンドの/Lスイッチや/Sスイッチです。

/Lスイッチのあとにある値は、UMBメモリの領域番号です。いくつの領域に分けられているかということを確認するには、次のように実行してください。

```
A:¥>MEM /F
```

デバイスドライバによっては、複数の領域が必要になります。その場合には、セミコロン（；）で区切ります。

領域番号のあとにカンマで区切られて指定されている値は、組み込み時に必要な最小メモリサイズです。プログラムをUMBメモリに組み込もうとする際、プログラムが組み込み時に必要とするメモリよりも、UMBメモリのほうが少ないと、そこで起動がストップしてしまうことがあります。この値を指定しておけば、それよりも大きなUMBメモリでなければ組み込みを試みなくなるため、ストップすることはなくなるのです。領域を複数指定している場合には、領域ごとに**最小メモリ**を指定することができます。

/Sスイッチは、組み込み時にUMBメモリを最小メモリまで縮小して、メモリを効率よく使用できるようにするための指定です。必ず/Lスイッチと一緒に用いられます。

MemMakerが決めた/Lスイッチや/Sスイッチの設定に関しては、勝手に値を変更するようなことはしないでください。ただし、次のように手作業で通常の設定に書き換えることは可能です。

```
DEVICEHIGH /L:1,12560 =A:¥DOS¥SETVER.EXE
              ↓
DEVICEHIGH=A:¥DOS¥SETVER.EXE
```

MemMaker 実行後の AUTOEXEC.BAT

AUTOEXEC.BAT のほうも、比較してみましょう。

◉ MemMaker 実行前の AUTOEXEC.BAT

```
@ECHO OFF
LH A:¥DOS¥SMARTDRV.EXE 3072
PATH A:¥JSLIB;A:¥WINDOWS;A:¥DOS;A:¥;A:¥TOOL
LH SHARE
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
REM CALL WINA
REM MENU
REM CD ¥
REM A:¥DOS¥SMARTDRV /C
```

◉ MemMaker 実行後の AUTOEXEC.BAT

```
@ECHO OFF
LH /L:0;2,16400 /S A:¥DOS¥SMARTDRV.EXE 3072
PATH A:¥JSLIB;A:¥WINDOWS;A:¥DOS;A:¥;A:¥TOOL
LH /L:2,15232 SHARE
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
REM CALL WINA
REM MENU
REM CD ¥
REM A:¥DOS¥SMARTDRV /C
```

LOADHIGH (LH) コマンドに対して、/L スイッチや/S スイッチが付けられています。それぞれの意味は、DEVICEHIGH コマンドの場合とまったく同じです。

Windows が起動されるようになっていた部分は、すべてコメントにしてありますが、MemMaker による最適化が完了すれば、これらの部分は元に戻してもだいじょうぶです。

MemMaker 実行後のメモリの様子

MemMaker を実行する前とあとの MEM コマンドの実行結果を比べてみましょう。

◉ MemMaker 実行前

```
メモリの種類          合計   =  使用   +  空き
----------------  -------  -------  -------
ｺﾝﾍﾞﾝｼｮﾅﾙ             640K      82K     558K
ｱｯﾊﾟｰ ﾒﾓﾘ              27K      13K      14K
予約済み                0K       0K       0K
XMS ﾒﾓﾘ           18,406K   6,678K  11,728K
----------------  -------  -------  -------
全ﾒﾓﾘ             19,072K   6,772K  12,300K

全 1MB 以下ﾒﾓﾘ        666K      94K     572K

全 EMS ﾒﾓﾘ                         18,368K  (18,808,832 ﾊﾞｲﾄ)
空き EMS ﾒﾓﾘ                       11,808K  (12,091,392 ﾊﾞｲﾄ)
                           この数字に注目
最大実行可能ﾌﾟﾛｸﾞﾗﾑｻｲｽﾞ            [558K]  (571,552 ﾊﾞｲﾄ)
最大空きｱｯﾊﾟｰﾒﾓﾘﾌﾞﾛｯｸ                 14K  (14,112 ﾊﾞｲﾄ)
MS-DOS はハイメモリ領域に常駐しています.
```

◉ MemMaker 実行後

```
メモリの種類          合計   =  使用   +  空き
----------------  -------  -------  -------
ｺﾝﾍﾞﾝｼｮﾅﾙ             640K      57K     583K
ｱｯﾊﾟｰ ﾒﾓﾘ              70K      37K      33K
予約済み                0K       0K       0K
XMS ﾒﾓﾘ           18,362K   6,682K  11,680K
----------------  -------  -------  -------
全ﾒﾓﾘ             19,072K   6,776K  12,296K

全 1MB 以下ﾒﾓﾘ        710K      94K     616K

全 EMS ﾒﾓﾘ                         18,368K  (18,808,832 ﾊﾞｲﾄ)
空き EMS ﾒﾓﾘ                       11,760K  (12,042,240 ﾊﾞｲﾄ)
                             増えている！
最大実行可能ﾌﾟﾛｸﾞﾗﾑｻｲｽﾞ            [583K]  (596,736 ﾊﾞｲﾄ)
最大空きｱｯﾊﾟｰﾒﾓﾘﾌﾞﾛｯｸ                 20K  (20,064 ﾊﾞｲﾄ)
MS-DOS はハイメモリ領域に常駐しています.
```

Step 49

SCSIインターフェイス使用時の裏ワザ

●ポイント●

- SCSIインターフェイスを用いている場合、EMM386.EXEに/MOVEHDBIOSスイッチを用いるとUMBメモリが増える。
- /MOVEHDBIOSスイッチを用いた箇所は、/UMBスイッチだけでは認識されないことがあるため、/Iスイッチを用いて強制的にUMBメモリにする。
- PCIバスでは/MOVEHDBIOSスイッチは無効になる。
- IDEインターフェイスのBIOSは移動できない。
- EMM386.EXEに特にオプションを指定している場合には、MemMakerで「現在のEMM386メモリ設定をこのまま保持しますか?」を「はい」にする。

●解 説●

SCSIインターフェイス使用時の設定

CD-ROM装置やMO(光ディスク)装置を接続する場合には、たいてい**SCSIインターフェイス**を用いることになると思います。しかし、SCSIインターフェイスボードを組み込むと、通常はUMBメモリとして利用されるアドレスDC000h〜DDFFFhの範囲が、ボードのROMアドレスとして占有されてしまうのです。

その結果、UMBメモリの容量が8Kバイト分減ってしまうだけではなく、通常は1つのブロックとして利用できるはずのアドレスD0000h〜DFFFFhのUMBメモリが、D0000h〜DBFFFhのブロックと、DE000h〜DFFFFhのブロックに分断されてしまい、大きなプログラムを組み込むことができなくなってしまいます。

◉ SCSIインターフェイスを使用した場合のメモリの様子

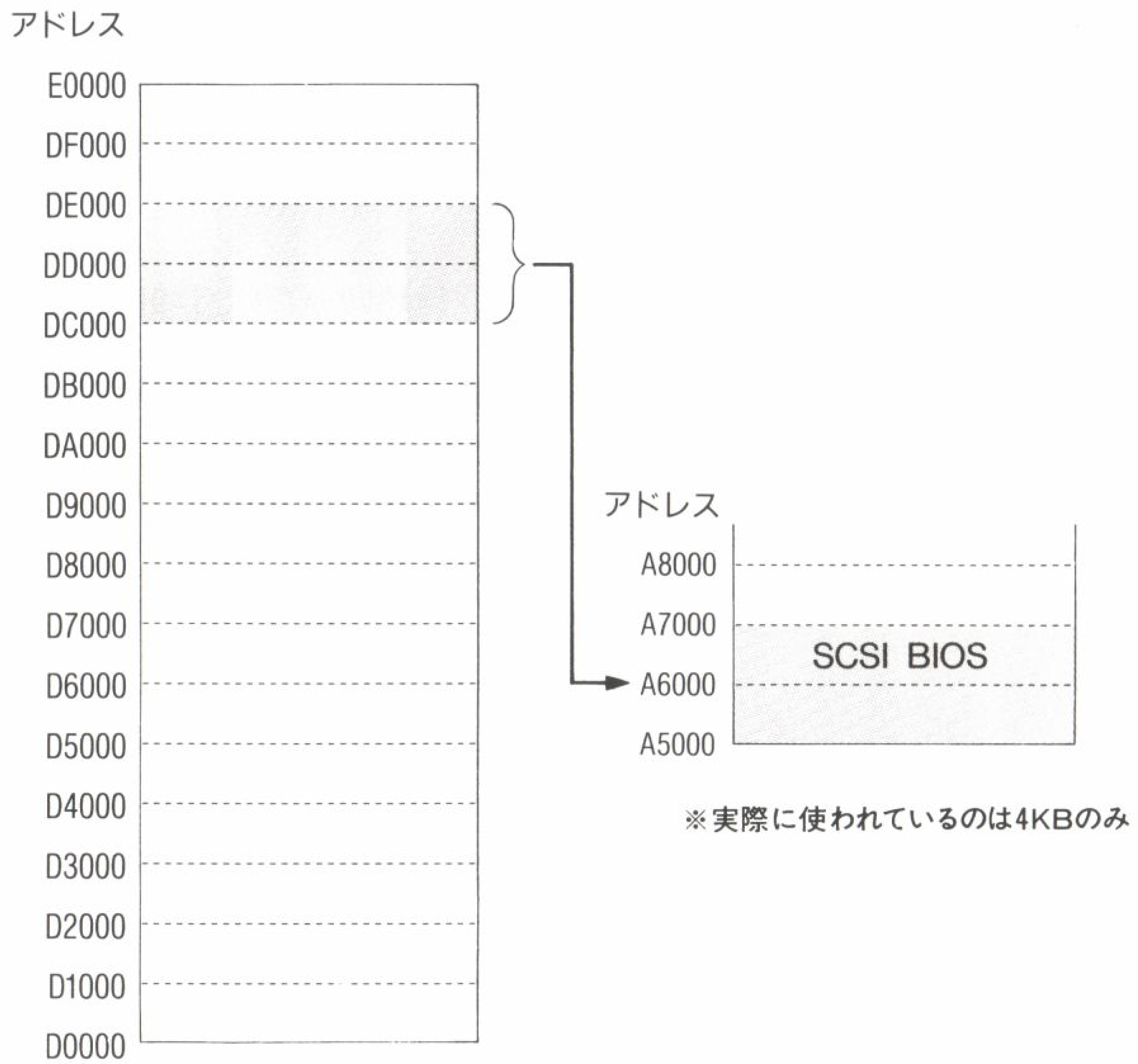


この問題を解決するには、EMM386.EXE に/**MOVEHDBIOS** という隠しスイッチを付けるという方法があります。

◉ EMM386.EXE の/MOVEHDBIOS スイッチを利用する

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB /MOVEHDBIOS /I=D000-DFFF /HIGHSCAN
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥DOS¥RAMDISK.SYS 6144
```

◉/MOVEHDBIOS スイッチを用いた場合のメモリの様子



/MOVEHDBIOSスイッチを付けると、アドレスDC000h～DDFFFhになっているSCSIインターフェイスボードおよび**SASIインターフェイスボード**（旧型の機種で用いられていた）のROMアドレスを、A5000h～A7FFFhの範囲に移動してくれるため、元通りのUMBメモリを確保できるようになるのです。

ただ、/MOVEHDBIOSスイッチだけではUMBメモリとして割り当てられなかったりするため、/Iスイッチを用いて強制的にUMBメモリとして割り当てるようにしましょう。この/Iスイッチは、アドレスを誤って指定するとシステムが起動できなくなることもあるため、慎重に設定してください。

PCIバス使用時の設定

PCIバスを用いたシステムの場合には、SCSIインターフェイスであっても/MOVEHDBIOSスイッチで移動させることはできません。

PCIバスのシステムでは、通常の設定でDC000h～DFFFFhまでがUMBメモリから除外されるはずですが、この一部はUMBメモリとして利用できるようです。

筆者の使用しているPC-9821Xa/C10Wでは、次のように/Iスイッチを用いてUMBメモリを確保しています。

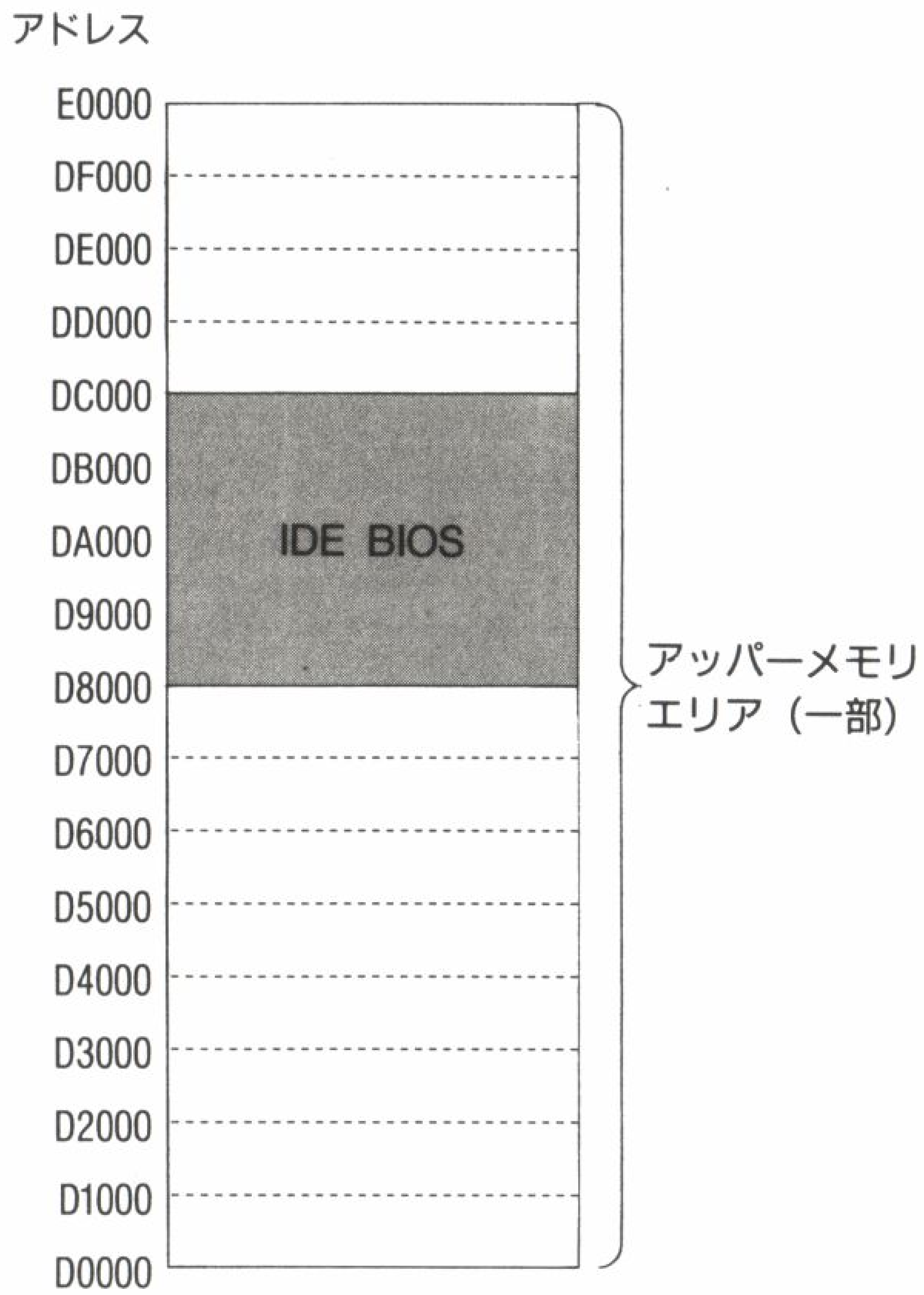
```
DEVICE=A:¥DOS¥EMM386.EXE /UMB /HIGHSCAN /I=A500-A7FF /I=DF00-DFFF
```

DF000h～DFFFFhまでは、実際には使われていないらしく、強制的にUMBメモリにしてしまっても、特に問題は起こっていません。また、/MOVEHDBIOSスイッチを用いないので、A5000h～A7FFFhまでをUMBメモリとして利用しています。

IDE インターフェイス使用時の設定

比較的新しい機種では、内蔵型のハードディスクが IDE という仕様のインターフェイスで接続されていることが多くなっています。特に、ノート型パソコンでは、ほとんどの機種が **IDE インターフェイス**のハードディスクを用いています。その場合、アドレス D8000h～DBFFFh の範囲は IDE の BIOS などが利用するため、UMB メモリとして使用することはできません。

◉ IDE インターフェイスが用いられている場合のメモリの様子



IDE インターフェイスに対しては、/MOVEHDBIOS スイッチを用いても無駄です。

ただし、A5000h～A7FFFh の範囲が利用できるため、EMM386.EXE の/I スイッチで指定しておくようにしましょう。

◉ A5000h〜A7FFFh を UMB にする

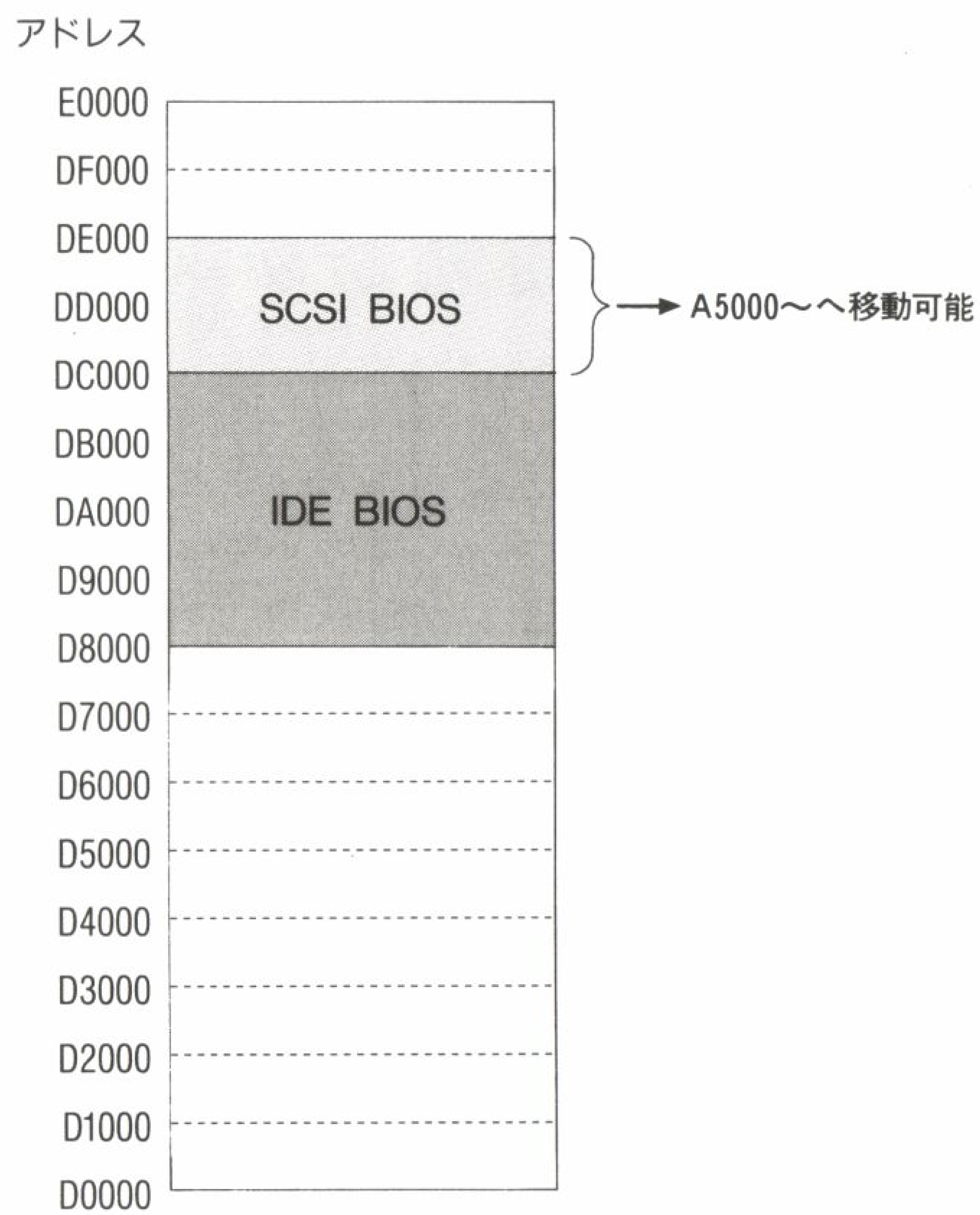
```
DEVICE=A:¥DOS¥EMM386.EXE /UMB /I=A500-A7FF /HIGHSCAN
```

これで、12K バイト分の UMB メモリが増えます。

IDE インターフェイスと SCSI インターフェイス使用時の設定

IDE インターフェイスを用いたシステムでは、MO などの周辺装置を接続するためには、さらに SCSI インターフェイスを用いなければならなくなります。そうなると、UMB メモリは極端に少なくなってしまいますが、どうしようもありません。

◉ IDE インターフェイスと SCSI インターフェイスを用いた場合のメモリ



その場合には、せめて/MOVEHDBIOSスイッチによって、SCSIのBIOSだけは移動しておくようにしましょう。

●/MOVEHDBIOSスイッチでSCSIのBIOSを移動する

```
DEVICE=A:¥DOS¥EMM386.EXE /UMB /MOVEHDBIOS /I=DC00-DFFF /HIGHSCAN
```

このケースでも、PCIバスを用いたシステムでは、/MOVEHDBIOSスイッチは使用できません。

UMBメモリを指定した場合のMemMaker

EMM386.EXEに、/MOVEHDBIOSスイッチや/Iスイッチを指定してある場合には、MemMakerの設定で「現在のEMM386メモリ設定をこのまま保持しますか?」を「はい」にしておいてください。
「上位メモリ領域を積極的にスキャンしますか?」のところは、「はい」にしておくと/HIGHSCANスイッチが付けられて、UMBメモリが広がります。
ただし、「上位メモリ領域をWindowsで使うために最適化しますか?」の問いには、UMBメモリに余裕がないシステムでは「いいえ」にしておくようにしてください。IDEとSCSIを両方とも使っているようなシステムでは、ほとんどUMBメモリがなくなってしまうからです。

Part 4
Other Tools編

Other Tools 編の内容

このOther Tools 編では、DOS6に付属のプログラム以外のものを扱っています。

まず最初に、市販の拡張メモリマネージャ『MEMORY SERVER II』と『MELWARE for Windows』の設定方法について解説します。

続いて、本書の付属ディスクに収められているフリーソフトウェアについての解説を行います。ここで解説するフリーソフトウェアを用いれば、DOSの環境がかなり改善されてくるはずです。設定が難しいものもありますが、あせらずじっくり取り組んでください。

Step 50

MEMORY SERVER IIを使用する

●ポイント●

- 『MEMORY SERVER II』がMS-DOS Ver.6.2に対応したのはバージョン1.16から。
- 『MEMORY SERVER II』では、HIMEM.SYS/EMM386.EXEの代わりにVMM386.EXEを用いる。
- PC-9821Cf/Cx/Cbでは、ディスクキャッシュプログラムDC10.EXEは使えない。
- ディスクキャッシュプログラムDC10.EXEよりSMARTDRV.EXEのほうが高機能。
- ディスクキャッシュドライバDC10.EXEは、256～16384の範囲から、EMSメモリなら16Kバイト単位、XMSメモリなら8Kバイト単位で容量を指定する。
- RAMディスクドライバIOS10.EXEは、128～32768の範囲から128Kバイト単位で容量を指定する。
- RAMディスクドライバIOS10.EXEを複数組み込むことにより、最大8台までのRAMディスクを作成できる。
- VMM386.EXEでは、/MスイッチでIDE BIOSが移動できる（制限あり）。
- システムROMにUMBメモリを割り当てている場合で、NEC版のWindowsを使用するためには、VMM386.EXEに/NECIDスイッチを付ける。
- メモリの最適化にはOPTUMB.EXEを用いる。
- DEVICEHIGHコマンドとLUMB.EXEは併用しない。

●解 説●

MEMORY SERVER IIの特徴

アイ・オー・データ機器のメモリ製品を購入すると、たいていは『MEMORY SERVER II』が添付されているはずです。この『MEMORY SERVER II』は、DOSに付属のHIMEM.SYSやEMM386.EXEの代わりに用いる拡張メモリマネージャと考えてよいのですが、DOS6にはない次のような特徴をもっています。

- SASI/SCSIインターフェイスのBIOSだけではなく、IDEのBIOSも移動できる。
- Windowsの常設スワップファイルにもキャッシング可能で、割り当て容量を起動後に変更可能なディスクキャッシュドライバが付属。
- 割り当て容量を起動後に変更可能なRAMディスクドライバが付属。
- UMBメモリを利用してディスクバッファの拡張が可能。
- Windows内からディスクキャッシュやRAMディスクの容量を変更できるデバイスコントロールが付属。

『MEMORY SERVER II』を利用する上で、最も大きなメリットとなりそうなのは、IDEインターフェイスBIOSの移動機能になると思います。ただし、ノートパソコンやPCIバスを使用したシステムなどでは移動できません。

なお、『MEMORY SERVER II』がMS-DOS Ver.6.2に対応したのは95年1月です。対応版の箱には、「対応OSの追加について:NEC製/EPSON製MS-DOS Ver.6.2」という表示がありますから、旧バージョンのユーザーはバージョンアップサービスを受けるようにしてください。本書の執筆に用いたファイルのバージョンは次のとおりです(バージョンはタイムスタンプの時刻でわかります)。

- VMM386.EXE → 3.12
- DC10.EXE → 3.34
- IOS10.EXE → 3.15

『MEMORY SERVER II』単体のほか、CONFIG.SYSやAUTOEXEC.BATを選択して高速な起動が可能な『REBOOT』と、Windows用の便利なユーティリティ『WinKit』がセットになった『SERVER II EX』もあります。

拡張メモリマネージャ VMM386.EXE のオプション

『MEMORY SERVER II』で用いられている **VMM386.EXE** は、DOS6 に付属の HIMEM.SYS と EMM386.EXE の機能を兼ね備えた拡張メモリマネージャです。常駐サイズが小さいので、このドライバと交換するだけでもメモリの節約になります。

VMM386.EXE で用いるオプションを、DOS6 に付属の EMM386.EXE のオプションと対比させてみましょう。ただし、初期状態で指定されたことになっているオプションは省いてあります。

◉ VMM386.EXE と EMM386.EXE のオプション

オプション	意　味	EMM386.EXE の該当オプション
/F = nn	EMS ページフレームの先頭アドレス	/F = nnnn(-mmmm)
/W = nn(-mm)	強制的に EMS ページフレームに割り当てるアドレス	なし
/U(= nn-mm)	UMB を利用可能にする(UMB に割り当てる領域)	/UMB(/I = nnnn-mmmm)
/ROM = nn(-mm)	ページフレームや UMB から除外する領域（ノーマルモード）	/E = nnnn-mmmm または /EN = nnnn-mmmm
/HROM = nn(-mm)	ページフレームや UMB から除外する領域（ハイレゾリューションモード）	/E = nnnn-mmmm または /EH = nnnn-mmmm
/@= EMM386.SYS	メインメモリが 768K バイトのとき、EMM386.SYS を使用した場合と同じメモリ環境を提供する(ハイレゾリューションモード)	なし
/M(= hh:nn-mm)	ハードディスクの BIOS ROM を移動する	/MOVEHDBIOS（SASI/SCSI のみ）
/NECID	BIOS ROM の NEC ID をエミュレートする	なし
/CA	サイリックス製 CPU のキャッシュを有効にする	なし
/I	起動時に詳細な情報を表示する	/VERBOSE
/T =ドライバ名	拡張タスクスワップドライバを組み込む（通常は EXTDSWAP.SYS)	/T = EXTDSWAP.SYS

オプション	意　味	EMM386.EXE の該当オプション
/NV	VCPI を使用しない	なし
/XH = nn	XMS のハンドル数	（HIMEM.SYS の/NUMHANDLES = nnn）
/EH = nnn	EMS のハンドル数	/H = n
/NE	EMS を使用しない（Windows を使用するなら指定してはいけない）	/NOEMS

EMS を使用しないようにする/NE スイッチは、EMM386.EXE の/NOEMS スイッチとは異なり、指定してしまうと Windows を起動できなくなるため、ほとんど使われることのないオプションと考えてよいと思います。

EMM386.EXE にはあって、VMM386.EXE にはないオプションとしては、主に次のものがあります。

- /M, /P　：EMS の容量を指定。
- /NOHI　：EMM386.EXE のコード部分を UMB に移動しない。
- /A = n　：高速代替マップレジスタ数。
- /DPMI　：DPMI 環境を使用できるようにする。
- /HIGHSCAN　：ROM BASIC が使用する領域を自動的に UMB に割り当てる。
- /WIN = nnnn-mmmm　：Windows が DOS 環境のアプリケーションを実行する際に使う変換バッファのために UMB を予約する。

VMM386.EXE では、初期状態で VCPI と DPMI が利用できる状態になります。そのため、EMM386.EXE にある/DPMI スイッチのようなオプションは必要ないのです。

なお、日本語 FEP として ATOK7 を利用する場合、Windows を起動するには VMM386.EXE に/ATOK7 スイッチを付ける必要があります。ただし、EMS の実行速度が低下することがあるため、できれば ATOK7 の使用は避けるほうがよいと思います。

ディスクキャッシュプログラム DC10.EXE のオプション

『MEMORY SERVER II』に付属のディスクキャッシュプログラム **DC10.EXE** でも、DOS6 に付属の SMARTDRV.EXE とオプションの比較をしてみましょう。やはり、初期状態で指定されたことになっているオプションは省いてあります。

● DC10.EXE と EMM386.EXE のオプション

オプション	意　味	SMARTDRV.EXEの該当オプション
/E	EMS メモリを使用する	なし
/S	データ領域を EMS メモリに配置する	なし
/NV	Windows の常設スワップファイルへのキャッシングを禁止する	なし
/W = n	Windows 起動時のキャッシュサイズ	初期キャッシュサイズのあとに値で指定
/RE	EMS フレーム上に 8K バイトの先読みバッファを設定する	なし
/RU	UMB 上に 8K バイトの先読みバッファを設定する	なし
/RM	DC10.EXE 常駐部の後ろに 8K バイトの先読みバッファを設定する	/B:nnnn
/R	/RE →/RU →/RM の順番で自動的に先読みバッファを設定する（使用可能な領域がな ければ設定しない）	なし
/NA	フロッピーディスクの入れ替えが発生しても警告音を鳴らさない	なし
/NL	FAT 情報もキャッシュバッファから掃き出す	なし
/NF	キャッシュバッファがいっぱいになっても、キャッシュバッファの掃き出しを行わない	なし
/NC	対象ドライブに対してキャッシングを行わない	ドライブ名-

DC10.EXE と SMARTDRV.EXE は、基本機能がかなり異なっているため、このような比較はあまり意味がないかもしれません。

DC10.EXE は、書き込みキャッシュの機能がないため、筆者としてはSMARTDRV.EXE のほうをお勧めしたいと思います。ただ、書き込みキャッシュはうっかり電源を切ってしまうとディスクに保存すべき内容が消えてしまうことがあるため、書き込みキャッシュを利用しないなら、DC10.EXE のほうがよさそうです。

DC10.EXE では、256〜16384 の範囲から、EMS メモリを利用するなら 16K バイト単位、XMS メモリなら 8K バイト単位で容量を指定してください。

なお、PC－9821Cf/Cx/Cb では DC10.EXE は使用できません。必ずSMARTDRV.EXE のほうを用いるようにしてください。

RAM ディスクドライバ IOS10.EXE のオプション

『MEMORY SERVER II』に付属の RAM ディスクドライバ **IOS10.EXE** でも、DOS6 に付属の RAMDISK.SYS とオプションの比較をしてみましょう。やはり、初期状態で指定されたことになっているオプションは省いてあります。

◉ IOS10.EXE と EMM386.EXE のオプション

オプション	意　味	RAMDISK.SYS の該当オプション
/E	EMS メモリを使用する	/E
/X	XMS メモリを使用する	なし（初期設定）
/R = d	指定したドライブ名を割り当てる	なし
/M	DOS シェル使用時に RAM ディスクとして認識されるようにする	なし
/NS	データチェックを行わない	なし
/C = nn	クラスタサイズをセクタ数で指定する	なし
/L = nnnn	セクタサイズを指定(256/512/1024)	容量のあとに値で指定(512/1024)
/D = nnnn	ルートディレクトリのエントリ数	セクタサイズのあとに値で指定
/A = yyxxz	アクセスランプの位置と色を指定	なし
/W	ライトプロテクトをかける	なし

IOS10.EXEは、ドライブ名の指定や、アクセスランプ(RAMディスクの使用中に表示が出る）の指定ができるほか、起動後に容量を変更する機能などもあるため、RAMDISK.SYSより高機能と言えるでしょう。

IOS10.EXEでは、128〜32768の範囲から128Kバイト単位で容量を指定してください。ただし、セクタサイズに256を指定した場合に限り、最大容量は16384(Kバイト)になります。

複数のIOS10.EXEを組み込むことにより、最大で8台までのRAMディスクを作成することができます。

『MEMORY SERVER II』の基本的な設定方法

『MEMORY SERVER II』の設定方法としては、付属のインストールプログラムを用いるのが一番簡単です。本書では、手作業でも設定の変更ができるように、CONFIG.SYSやAUTOEXEC.BATでの具体的な記述方法について解説しておきます。

なお、『MEMORY SERVER II』にもDOS6のMemMakerと同じ最適化を行ってくれる**OPTUMB.EXE**が付属になっているため、これからの解説ではデバイスドライバや常駐プログラムのUMBメモリへの読み込みは指定していません。あとで、OPTUMB.EXEを実行するものとします。

まず、DOS6で次のような基本設定を作成した状態から考えてみましょう。

◉ CONFIG.SYS

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥DOS¥RAMDISK.SYS 6144
```

◉ AUTOEXEC.BAT

```
@ECHO OFF
PATH=A:¥;A:¥DOS
SMARTDRV 3072
SET DOSDIR=A:¥DOS
SET TEMP=E:¥
PROMPT $P$G
```

同じ内容を『MEMORY SERVER II』で設定した場合、次のようになります。

◉ CONFIG.SYS

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U
DEVICE=A:¥MDEV¥IOSPRO¥DC10.EXE 3072
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥MDEV¥IOSPRO¥IOS10.EXE 6144 /X
```

◉ AUTOEXEC.BAT

```
@ECHO OFF
PATH=A:¥;A:¥DOS
SET DOSDIR=A:¥DOS
SET TEMP=E:¥
PROMPT $P$G
```

基本的には、拡張メモリマネージャ、ディスクキャッシュプログラム、RAMディスクドライバを入れ替えるだけと考えてください。CONFIG.SYSの基本コマンドについては、DOS6に付属のドライバ類を用いる場合と同様に設定します。

DC10.EXEの設定

『MEMORY SERVER II』では、ディスクキャッシュプログラムDC10.EXEをCONFIG.SYS内で設定することができます。SMARTDRV.EXEのように、AUTOEXEC.BAT内で実行することもできますが、なるべく早く確保しておくほうが起動時間が若干短縮されるはずです。

DC10.EXEは、ドライブを指定しないで組み込むと、すべてのハードディスクに対してキャッシングが行われます。光ディスクのドライブがあれば、次のようにドライブ名を指定するとキャッシングが有効になります。

```
DEVICE=A:¥MDEV¥IOSPRO¥DC10.EXE 3072 D:
```

ただし、CD-ROMドライブに対してキャッシングを行うことはできません。

DC10.EXEの代わりにSMARTDRV.EXEを使うと、次のようになります。

◉ CONFIG.SYS

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥MDEV¥IOSPRO¥IOS10.EXE 6144 /X
```

◉ AUTOEXEC.BAT

```
@ECHO OFF
PATH=A:¥;A:¥DOS
SMARTDRV 3072
SET DOSDIR=A:¥DOS
SET TEMP=E:¥
PROMPT $P$G
```

書き込みキャッシュの機能があることと、CD-ROMドライブまでキャッシングが可能なことを考えると、SMARTDRV.EXEを使うほうがよさそうです。

IOS10.EXEの設定

IOS10.EXEは、特に指定をしないとEMSメモリから割り当てられますから、XMSメモリから割り当てられるようにするなら、/Xスイッチを付けてください。

また、ハードディスクのように、読み書きの際にランプが点灯する機能を使いたければ、/Aスイッチを付けます。次のようなパラメータで、アクセスランプとしてマークを表示する位置と、マークの色を指定してください。

◉ IOS10.EXEの/Aスイッチ

- **書式**：/A=yyxxz
- yy ：アクセスランプの行位置
  20行モードの場合：00〜19（FNにするとファンクション表示の行）
  25行モードの場合：00〜24（FNにするとファンクション表示の行）
- xx ：アクセスランプの桁位置
  00〜79
- z ：アクセスランプの色
  0:黒　1:赤　2:緑　3:黄色　4:青　5:紫　6:水色　7:白

たとえば、ファンクションライン（最下位行）の左端に赤色で表示するなら、次のように指定します。yyとxxは、必ず2桁で指定してください。

```
DEVICE=A:¥MDEV¥IOSPRO¥IOS10.EXE 6144 /X /A=FN001
```

特に大きなRAMディスクを作成する場合には、システム空間（アドレスF00000h〜FFFFFFh）による拡張メモリの分断に配慮してください。XMSメモリから割り当てた場合、確保できる最大サイズは連続している最大の拡張メモリの範囲になります。大容量のRAMディスクを作成するなら、/Xスイッチを付けずにEMSメモリから割り当てるほうがよさそうです。

VMM386.EXEのケース別の設定方法

今度は、いくつかのケースに分けてVMM386.EXEの設定方法について解説します。

サウンドBIOSの部分にUMBメモリを割り当てる

FM音源を搭載した機種では、アドレスCC000h〜CDFFFhの範囲をBIOSが使用するため、そのままではEMSのページフレームがうまく設定できません。その場合には、HELPキーを押しながらシステムを起動した際に表示されるメニューから、BIOSを切り離すように設定すればよいのですが、VMM386.EXEならもっと簡単に対処することが可能です。

サウンドBIOSの領域に強制的にUMBメモリを割り当てるため、/Wスイッチを用いてください。また、/Fスイッチでページフレームの開始アドレスを指定しておきます。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /F=C0 /W=CC /U
```

/Fスイッチの「C0」はアドレスC0000hを、/Wスイッチの「CC」はCC000hをそれぞれ指しています。

これで、通常通りC0000hからページフレームが設定されます。

システムROMなどにUMBメモリを割り当てる

DOS6に付属のEMM386.EXEには、システムROMとして使われていることになっているE8000h〜F3FFFhの範囲に、UMBメモリを割り当てる/HIGHSCANスイッチがあります。これと同じことをVMM386.EXEで行うには、/Uスイッチでアドレスを指定してください。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U=D0-D7,DC-DF,E8-F3 （IDE使用の場合）
```

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U=D0-DB,DE-DF,E8-F3 （SCSI使用の場合）
```

この場合には、UMBメモリを割り当てるすべての領域をアドレスで指定しなければならないため、ちょっと複雑な設定になります。

VMM386.EXEにおけるアドレスの指定は、4Kバイト単位で次のように行います。

```
・D0           →  D0000h～D0FFFh
・D0-D7        →  D0000h～D7FFFh
・DE-DF,E8-F3 →  DE000h～DFFFFh と E8000h～F3FFFh
```

テキストVRAMの未使用部分(A5000h～A7FFFh)にも、UMBメモリを割り当てることができます。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U=A5-A7,D0-D7,DC-DF,E8-F3
```

筆者が使用しているPC-9821Xa/C10Wのように、PCIバスを搭載したシステムでは、IDE BIOSと合わせてD8000h～DFFFFhまでがUMBメモリとして利用できなくなっています。しかし、そのうちのDF000h～DFFFFhまではUMBメモリを割り当てても問題はなさそうです。そこで、筆者は次のように設定しています。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U=A5-A7,D0-D7,DF,E8-F3
```

SASI/SCSI/IDEインターフェイスのBIOSを移動する

EMM386.EXEの/MOVEHDBIOSにあたるスイッチとして、VMM386.EXEには/Mが用意されています。VMM386.EXEでは、IDEインターフェイスのBIOSまで移動することができます(ノートパソコンの場合やPCIバスを用いている場合などを除く)。

ただし、次の場合には、IDE BIOSを移動できませんから注意してください。

①ノート型パソコンの場合（IDEのBIOS以外のBIOSが存在するため)。
② PCIバス搭載のシステムの場合。
③ Plug&Play機能を使う場合(CONFIG.SYSに、DEVICE = A:¥PLUGPLAY¥DRIVERS¥DOS¥DWCFGMG.SYSの記述がある場合など)。
④ EPSONのデスクトップ機。
⑤ IDEのROM領域を直接参照するドライバを使用する場合。

SASIおよびSCSIインターフェイスのBIOSを移動するには、VMM386.EXEに/Mスイッチを付けてください。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U /M
```

これで、D6000h〜D7FFFh の SASI BIOS と、DC000h〜DDFFFh の SCSI BIOS が、A5000h〜A7FFFh の領域に移動されます。それぞれの領域を足すと 16K バイトになって、A5000h〜A7FFFh の 12K バイトの範囲に収まらないように思えますが、実際に使われているのはそれぞれ 4K バイトだけなので、移動後には収まるようになっているのです。

● SCSI BIOS だけの移動

SCSI BIOS だけを移動するなら、次のようにしてください。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U /M=DC:8
```

「=」のあとにある「DC」が、移動したい BIOS の先頭アドレスを指しています（ここでは DC000h)。「：」のあとの「8」は、移動する領域の大きさになります（ここでは 8K バイト)。

● IDE BIOS だけの移動

IDE BIOS の場合は、サイズが 16K バイトもあるため A5000h〜A7FFFh（12K バイト）の領域には移動できません。そこで、どこか適当なブロックの最後の部分か、または最初の部分に移動してしまいましょう。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U=A5-A7,D0-DF,EC-F3 /M=D8:16-E8
```

この例では、システム ROM の E8000h〜EBFFFh の範囲に移動して、残りのシステム ROM 領域と A5000h〜A7FFFh の領域には UMB メモリを割り当てています。

もしも、SCSI と IDE の両方を使用しているなら、次のように設定するとよいでしょう。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U=D0-DF,EC-F3 /M=D8:16-E8,DC:8
```

Windows を使用する(システム ROM に UMB メモリを割り当てた場合)

システム ROM に UMB メモリを割り当ててしまうと、NEC 版の Windows が起動できなくなる場合があります。その際には、/NECID スイッチを付けてください。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U=D0-D7,DC-DF,E8-F3 /NECID
```

Microsoft 版の Windows では、このスイッチは必要ありません。

/NECID スイッチは、NECCD.SYS や N88BASIC を使用する際にも指定しておいてください。

DPMI 環境を利用する

『一太郎 Ver.5』のように、DPMI 環境で動作するアプリケーション(DPMI クライアント)を利用するためには、DPMI サーバとして **DPMI32.EXE** を組み込む必要があります。

◉ DPMI32.EXE の組み込み

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U=A5-A7,D0-D7,DC-DF,E8-F3
DEVICE=A:¥MDEV¥IOSPRO¥DPMI32.EXE
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥MDEV¥IOSPRO¥IOS10.EXE 6144 /X
```

このように設定すると、DPMI サーバは UMB メモリのうちの 4K バイトをストリームバッファとして利用します。

CONFIG.SYS で組み込む代わりにコマンドとして実行することもできます。

```
A:¥>DPMI32
```

なお、DPMI32.EXE を組み込んだ状態では、DOS シェルは使用できません。

OPTUMB.EXEによる最適化

『MEMORY SERVER II』において、DOS6のMemMakerにあたるのが**OPTUMB.EXE**です。CONFIG.SYSとAUTOEXEC.BATの設定が終わったら、最後にOPTUMB.EXEによって最適化を行うようにしましょう。

OPTUMB.EXEには、UMB領域を指定したり、SASI/SCSI/IDE BIOSの移動を指定する機能もありますが、これまでに解説した内容を参考にして、VMM386.EXEのスイッチ類はすべて手作業で指定しておくほうが効率がよいはずです。

最適化を行う前に、AUTOEXEC.BATの最後でWindowsやアプリケーションを起動するようにしてある場合には、そこをコメントにしておきましょう。『MEMORY SERVER II』をインストールしたディレクトリにパスが通っていれば、OPTUMB.EXEは次のように実行します。

```
A:¥>OPTUMB
```

パスを通していなければ、あらかじめディレクトリを移動しておいてください。

OPTUMB.EXEの設定方法

OPTUMB.EXEには、次の3種類の設定方法があります。

① UMB空き領域設定

メモリの様子がマップで表示されるので、そのマップを見ながらUMBメモリを割り当てたり、ハードディスクのBIOSを移動します

② UMB最適化設定

現在のCONFIG.SYSやAUTOEXEC.BATの内容に従って、なるべく多くのデバイスドライバや常駐プログラムがUMBメモリに収まるようにします

③ UMB空き領域・UMB最適化同時設定

①と②の作業を同時に行います

VMM386.EXEの指定が済んでいれば、②を実行してください。

ほとんど[⏎]キーを押しているだけで作業が進められていきますが、途中で、CONFIG.SYSとAUTOEXEC.BATのなかから、UMBメモリに組み込みたい行をマークするところがあります。スペースキーを使ってマークしてください。

最適化が終了すると、次のようなCONFIG.SYSやAUTOEXEC.BATになっているはずです。

◉最適化後のCONFIG.SYS

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /U=D0-D7,DF,A5-A7,E8-F3
BUFFERS=6
FILES=30
DOS=HIGH,UMB
LASTDRIVE=F
SHELL=A:¥COMMAND.COM A:¥ /P
DEVICE=A:¥MDEV¥IOSPRO¥LUMB.EXE /M /B=1 A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥WX3¥WXK.SYS /A5
DEVICE=A:¥WX3¥WX3.SYS /INI=A:¥WX3¥WX3SYS.INI /A5
DEVICE=A:¥MDEV¥IOSPRO¥IOS10.EXE 6144 /X
```

◉最適化後のAUTOEXEC.BAT

```
@ECHO OFF
PATH A:¥WINDOWS;A:¥DOS;A:¥;A:¥MDEV¥IOSPRO
SET TEMP=E:¥
SET DOSDIR=A:¥DOS
A:¥MDEV¥IOSPRO¥LUMB.EXE /M A:¥DOS¥SMARTDRV.EXE 5120 3072
A:¥MDEV¥IOSPRO¥LUMB.EXE /M DOSKEY /INSERT
PROMPT $P$G
```

『MEMORY SERVER II』では、DEVICEHIGHやLHの代わりに**LUMB.EXE**を用いてUMBメモリへと読み込んでいます。

/Mスイッチは、UMBメモリに入らなかった場合メインメモリに読み込むようにするためのスイッチです。また、/Bスイッチでは、読み込むUMBメモリのブロックを指定しています。

なお、最適化前にDEVICEHIGHコマンドやLHコマンドが用いられている場合でも、その部分はLUMB.EXEに置き換えられるなどするため、最適化には問題ありません。ただし、手作業でDEVICEHIGHコマンドとLUMB.EXEを併用するような設定にしないよう注意してください。

ディスクキャッシュの容量を変更する

『MEMORY SERVER II』では、起動後にディスクキャッシュとRAMディスクの容量を変更することが可能です。ただし、DOS6のDoubleSpace機能を利用している場合には、容量の変更はできません。

DC10.EXEの割り当て容量を変更するには、/@=SIZEを用います。ただし、最初に確保した容量よりも大きな容量には変更できません。

◉ DC10.EXEの/@=SIZEパラメータ

/@=SIZE(n)	：通常時とWindows起動時のキャッシュサイズをnに変更。
/@=SIZE(n,m)	：通常時のキャッシュサイズをnに、Windows起動時のキャッシュサイズをmに変更。
/@=SIZE(n,)	：通常時のキャッシュサイズをnに変更、Windows起動時のキャッシュサイズは変更しない。
/@=SIZE(,n)	：Windows起動時のキャッシュサイズをnに変更、通常時のキャッシュサイズは変更しない。

nやmには、EMSメモリなら「16～最初に確保した容量」の範囲から16Kバイト単位で、XMSメモリなら「8～最初に確保した容量」の範囲から8Kバイト単位でそれぞれ指定してください。

たとえば、通常時の容量を2048Kバイトに、Windows起動時の容量を1024Kバイトに変更するなら、次のように実行します。

```
A:¥>DC10 /@=SIZE(2048,1024)
```

RAMディスクの容量を変更する

IOS10.EXEの割り当て容量を変更するには、/Yスイッチを用いて128Kバイト単位で指定します。EMSメモリまたはXMSメモリに余裕があれば、最初の容量よりも大きくすることも可能です。

容量の変更を指定すると、RAMディスク内のファイルはすべて消去されてしまいますから、保存したいファイルがあれば、容量を変更する前にハードディスクなどに保存しておくようにしてください。

たとえば、容量を3072Kバイトに変更するなら、次のように実行します。

```
A:¥>IOS10 /Y=3072
```

構成を変更するという確認がありますから、Yキーを押してください。Nキーで変更を中止します。

キャッシュするファイルを固定する

通常のディスクキャッシュの設定では、あとから読み込んだファイルによってキャッシュ内がいっぱいになると、古い内容から順に捨てられていきます。これを、「**掃き出し**」と呼んでいます。

しかし、複数のファイルをワープロソフトで編集するような場合、メインで編集したいファイルだけはキャッシュの掃き出しを禁止しておくと、途中で別のファイルを読み込んでもキャッシュの効果が失われず便利です。ただし、DOS6のDoubleSpace機能を利用している場合には、掃き出しの禁止はできません。

DC10.EXEで、掃き出しを禁止するには、次のように/@=LOCKFILEで指定します。

◉/LOCKFILEによる掃き出し禁止の指定

```
/@=LOCKFILE(A:¥WX3.DIC)
    → A:¥WX3.DICを掃き出し禁止にする
/@=LOCKFILE(B:¥DOC¥SAMPLE1.TXT,B:¥DOC¥SAMPLE2.TXT)
    → B:¥DOC¥SAMPLE1.TXTとB:¥DOC¥SAMPLE2.TXTを掃き出
      し禁止にする
/@=LOCKFILE(B:¥DOC¥*.TXT)
    → B:¥DOCディレクトリ内の拡張子がTXTのファイルをすべて掃き出
      し禁止にする
/@=LOCKFILE(A:¥DOC¥)
    → A:¥DOCディレクトリ内の全ファイルを掃き出し禁止にする
```

ファイル名は、カンマ(,)で区切って複数を指定できるほか、ワイルドカードの使用も可能です。

ディレクトリ単位で指定するには、最後に「¥」を付けてください。

たとえば、A:¥DOCディレクトリ内の全ファイルを掃き出し禁止にするなら、次のように実行します。

```
A:¥>DC10 /@=LOCKFILE(A:¥DOC¥)
```

これで、対象となるファイルを読み込んだ場合、ディスクへのアクセスは行われないはずです。もちろん、ファイルの総容量よりもキャッシュメモリの容量を大きくしておく必要があります。

ファイルの掃き出しを禁止した状態でも、キャッシュメモリに余裕があれば、その分だけは通常通りのキャッシュ機能がはたらきます。

ディスクバッファの拡張

DOS6では、CONFIG.SYSのBUFFERSコマンドで指定するディスクバッファのサイズを、HMAメモリ内に収めるほうがメモリの節約になります。しかし、もしもUMBメモリに余裕があれば、ディスクバッファに割り当てるとよいでしょう。

ディスクバッファの拡張は、**BEX.COM**によって行います。MAXオプションを付けると、確保可能な最大の容量に拡張されます。

◉ BEX.COMでディスクバッファを拡張する

```
A:¥>BEX MAX
```

バッファの個数を指定することもできますが、その場合には/Hスイッチを付けて、指定個数が確保できなければ、確保できる最大の個数に設定されるようにしておくとよいでしょう。

◉バッファの個数を指定する

```
A:¥>BEX 20 /H
```

なお、バッファを拡張したあとは、HMAメモリ内のバッファは無効となります。したがって、HMAメモリ内に6個分確保してあって、14個拡張したからといって、全部で20個分のバッファになるわけではありません。

Step 51

MELWARE for Windowsを使用する

●ポイント●

- 『MELWARE for Windows』のドライバ類より、DOS6に付属のドライバ類のほうが高機能。
- 『MELWARE for Windows』では、HIMEM.SYS/EMM386.EXEの代わりにMELEMM.386を用いる。
- RAMディスクドライバEXDISK.EXEは、常駐プログラムとして組み込むことができる。
- メモリの最適化にはOPTIMIZE.COMを用いる。
- OPTIMIZE.COMでは細かな指定ができないため、若干の手直しが必要な場合がある。

●解 説●

MELWARE for Windowsの特徴

メルコのメモリ製品を購入すると、たいていは『MELWARE for Windows』が添付されているはずです。従来の『MELWARE』は、『MEMORY SERVER II』と同様に、DOSに付属のHIMEM.SYSやEMM386.EXEの代わりに用いる拡張メモリマネージャでしたが、『MELWARE for Windows』になってからは、どちらかというとWindows用のユーティリティという意味合いが強くなったように感じられます。実際、本書の執筆時点のバージョン2では、どちらかというとDOS6に付属の拡張メモリマネージャのほうが優っている部分が多いのです。

『MELWARE for Windows』が優っていると言える機能は、次の2点です。

- **割り当て容量を起動後に変更可能なディスクキャッシュドライバが付属。**
- **割り当て容量を起動後に変更可能なほか、起動後にドライブの作成が可能なRAMディスクドライバが付属。**

いずれ、バージョンアップがあるかもしれませんが、現状ではWindows用ユーティリティのほうだけを利用するほうがよさそうに思えます。ただ、ディスクキャッシュプログラムがたいへん高速で、RAMディスクドライバがデバイスドライバとしてだけではなく、常駐プログラムとしても作成可能といった点は魅力的だと思います。Windows用のユーティリティは、たいへん便利なものが揃っていてお勧めです。

拡張メモリマネージャMELEMM.386のオプション

『MELWARE for Windows』で用いられている**MELEMM.386**は、DOS6に付属のHIMEM.SYSとEMM386.EXEの機能を兼ね備えた拡張メモリマネージャです。常駐サイズが小さいので、このドライバと交換するだけでもメモリの節約になります。

MELEMM.386で用いるオプションを、DOS6に付属のEMM386.EXEのオプションと対比させてみましょう。ただし、初期状態で指定されたことになっているオプションは省いてあります。

◉ MELEMM.386とEMM.386のオプション

オプション	意　味	EMM386.EXEの該当オプション
/SD nnnn	EMSページフレームの先頭アドレス	/F = nnnn(-mmmm)
/Pn	強制的にEMSページフレームに割り当てる（nは物理ページ数：通常は4を指定）	なし
/M nn,mm……	UMBを利用可能にする（UMBに割り当てる領域）	/UMB(/I = nnnn-mmmm)
/SWI	ハードディスクのBIOS ROMを移動する	/MOVEHDBIOS
/CX	サイリックス製CPUのキャッシュを有効にする	なし
/T	ドライバ名 拡張タスクスワップドライバを組み込む（通常はEXTDSWAP.SYS）	/T = EXTDSWAP.SYS
/H nnn	EMSのハンドル数	/H = n
/HM	XMSを有効にする	なし
/NC	EMSの一部のファンクションを高速化する	なし

EMM386.EXEにはあって、VMM386.EXEにはないオプションとしては、主に次のものがあります。

- /M, /P　：EMSの容量を指定。
- /NOHI　：EMM386.EXEのコード部分をUMBに移動しない。
- /A = n　：高速代替マップレジスタ数。
- /DPMI　：DPMI環境を使用できるようにする。
- /HIGHSCAN　：ROM BASICが使用する領域を自動的にUMBに割り当てる。
- /WIN = nnnn-mmmm　：WindowsがDOS環境のアプリケーションを実行する際に使う変換バッファのためにUMBを予約する。

MELEMM.386では、DPMIはサポートされていません。

ディスクキャッシュプログラムHYPERDSK.EXEのオプション

HYPERDSK.EXEは、DOS環境はもちろんのこと、特にWindows環境において威力を発揮するXMSメモリ専用の超高速ディスクキャッシュプログラムです。DOS6に付属のSMARTDRV.EXEとオプションの比較をしてみましょう。初期状態で指定されたことになっているオプションは省いてあります。

◉ HYPERDSK.EXEとSMARTDRV.EXEのオプション

オプション	意　味	SMARTDRV.EXEの該当オプション
C:n	DOSでのキャッシュサイズ	値で指定
CW:n	Windows起動時のキャッシュサイズ	初期キャッシュサイズのあとに値で指定
S	書き込みキャッシュを有効にする	なし（初期設定）
D	キャッシングを中止する	なし
T:n	書き込みキャッシュの遅延時間(nには0～30を指定：1につき約0.5秒)	なし

HYPERDSK.EXEは、DOS6のSMARTDRV.EXEと同様に、書き込みキャッシュの機能をもっています。ただし、デバイスドライバを経由してアクセスするメディア(CD-ROM、MO、PDなど)に対してはキャッシングできません。また、フロッピーディスクドライブも対象外となります。

HYPERDSK.EXEを利用する場合には、Windowsの「メイン」グループにある「コントロールパネル」から「エンハンスドモード」を実行して、「スワップファイルの設定」にある「BIOSを経由しないでスワップファイルを利用」の項目を確認してください。ここをチェック状態にしないように設定します。

なお、HYPERDSK.EXEは512バイト／セクタでフォーマットされているハードディスク以外では使用できません。

RAM ディスクドライバ EXDISK.EXE のオプション

RAM ディスクドライバ **EXDISK.EXE** でも、DOS6 に付属の RAMDISK.SYS とオプションの比較をしてみましょう。やはり、初期状態で指定されたことになっているオプションは省いてあります。

● EXDISK.EXE と RAMDISK.SYS のオプション

オプション	意　味	RAMDISK.SYS の該当オプション
E	EMS メモリを使用する	/E
/n	ルートディレクトリのエントリ数	セクタサイズのあとに値で指定

EXDISK.EXE では、基本的には容量の指定を行うだけになります。特に指定がなければ XMS メモリから割り当てられますが、E オプションを付けると EMS メモリが使われます。

ルートディレクトリのエントリ数は、32〜480 の範囲で指定してください。初期設定は 128 です。

EXDISK.EXE は、デバイスドライバとして組み込むだけではなく、常駐プログラムとしても実行できるので、かなり自由度が高いプログラムと言えます。

『MELWARE for Windows』の基本的な設定方法

『MELWARE for Windows』の設定方法としては、付属のインストールプログラムを用いるのが一番簡単です。本書では、手作業でも設定の変更ができるように、CONFIG.SYS や AUTOEXEC.BAT での具体的な記述方法について解説しておきます。

なお、『MELWARE for Windows』にも DOS6 の MemMaker と同じ最適化を行ってくれる **OPTIMIZE.COM** が付属になっているため、これからの解説ではデバイスドライバや常駐プログラムの UMB メモリへの読み込みは指定していません。あとで、OPTIMIZE.COM を実行するものとします。

まず、DOS6で次のような基本設定を作成した状態から考えてみましょう。

◉ CONFIG.SYS

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /UMB
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥DOS¥RAMDISK.SYS 6144
```

◉ AUTOEXEC.BAT

```
@ECHO OFF
PATH=A:¥;A:¥DOS
SMARTDRV 3072
SET DOSDIR=A:¥DOS
SET TEMP=E:¥
PROMPT $P$G
```

同じ内容を『MELWARE for Windows』で設定した場合、次のようになります。

◉ CONFIG.SYS

```
BUFFERS=6
FILES=30
SHELL=A:¥COMMAND.COM A:¥ /P
DOS=HIGH,UMB
DEVICE=A:¥MEL4WIN¥MELEMM.386 /M D0,D4,DC /HM
DEVICE=A:¥MEL4WIN¥HYPERDSK.EXE C:3072 S
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥MEL4WIN¥EXDISK.EXE 6144
```

◉ AUTOEXEC.BAT

```
@ECHO OFF
PATH=A:¥;A:¥DOS
SET DOSDIR=A:¥DOS
SET TEMP=E:¥
PROMPT $P$G
```

基本的には、拡張メモリマネージャ、ディスクキャッシュプログラム、RAMディスクドライバを入れ替えるだけと考えてください。CONFIG.SYSの基本コマンドについては、DOS6に付属のドライバ類を用いる場合と同様に設定します。

MELEMM.386の設定

MELEMM.386でUMBメモリとして割り当てる領域は、/Mスイッチのあとに16Kバイト単位で指定するようにしてください。16Kバイト単位ですから、それに満たないA5000h～A7FFFhのような領域は指定できません。たとえばD0を指定すると、D0000h～D3FFFhまでを設定することになります。

XMSを有効にするため、/HMスイッチは必須です。

/NCスイッチを付けると、EMSの速度が多少速くなります。

HYPERDSK.EXEの設定

『MELWARE for Windows』では、ディスクキャッシュプログラムHYPERDSK.EXEをCONFIG.SYS内で設定することができます。SMARTDRV.EXEのように、AUTOEXEC.BAT内で実行することもできますが、なるべく早く確保しておくほうが起動時間が若干短縮されるはずです。

HYPERDSK.EXEは、Sオプションを付けて組み込まないと書き込みキャッシュが有効になりません。

RAMディスクドライバEXDISK.EXEは、次のようにして常駐プログラムとして組み込むことも可能です。

```
EXDISK DEVICE=6144
```

このようにして組み込むと、あとからRAMディスクのメモリを解放することも可能になります。

◉ RAMディスクのメモリを解放する

```
A:¥>EXDISK R
```

ただし、この場合にはLASTDRIVEコマンドによって、あらかじめドライブを確保しておく必要があります。

MELEMM.386のケース別の設定方法

今度は、いくつかのケースに分けてMELEMM.386の設定方法について解説します。

システムROMなどにUMBメモリを割り当てる

DOS6に付属のEMM386.EXEには、システムROMとして使われていることになっているE8000h〜F3FFFhの範囲に、UMBメモリを割り当てる/HIGHSCANスイッチがあります。これと同じことをMELEMM.386で行うには、/Mスイッチでアドレスを指定してください。

```
DEVICE=A:¥MEL4WIN¥MELEMM.386 /M D0,D4,DC,E8,EC,F0 /HM (IDE使用の場合)
DEVICE=A:¥MEL4WIN¥MELEMM.386 /M D0,D4,D8,E8,EC,F0 /HM (SCSI使用の場合)
```

この場合には、UMBメモリを割り当てるすべての領域をアドレスで指定しなければならないため、ちょっと複雑な設定になります。MELEMM.386におけるアドレスの指定は、16Kバイト単位で次のように行います。

- D0 → D0000h〜D3FFFh
- D4 → D4000h〜D7FFFh
- D8 → D8000h〜DBFFFh
- DC → DC000h〜DFFFFh

16Kバイト未満の領域は、たとえ余っていてもUMBメモリを割り当てることができないのが残念です。

SASI/SCSIインターフェイスのBIOSを移動する

EMM386.EXEの/MOVEHDBIOSにあたるスイッチとして、MELEMM.386には/SW1スイッチが用意されています。

```
DEVICE=A:¥MEL4WIN¥MELEMM.386 /M D0,D4,D8,DC,E8,EC,F0 /HM /SW1
```

これで、D6000h～D7FFFhのSASI BIOSと、DC000h～DDFFFhのSCSI BIOSが、A5000h～A7FFFhの領域に移動されます。それぞれの領域を足すと16Kバイトになって、A5000h～A7FFFhの12Kバイトの範囲に収まらないように思えますが、実際に使われているのはそれぞれ4Kバイトだけなので、移動後には収まるようになっているのです。

OPTIMIZE.COMによる最適化

『MELWARE for Windows』において、DOS6のMemMakerにあたるのが**OPTIMIZE.COM**です。CONFIG.SYSとAUTOEXEC.BATの設定が終わったら、最後にOPTIMIZE.COMによって最適化を行うようにしましょう。ただし、カードサービス（CS.EXE）を使用している環境では実行しないでください。

最適化を行う前に、AUTOEXEC.BATの最後でWindowsやアプリケーションを起動するようにしてある場合には、そこをコメントにしておきましょう。『MELWARE for Windows』をインストールしたディレクトリにパスが通っていれば、OPTIMIZE.COMは次のように実行します。

```
A:¥>OPTIMIZE
```

パスを通していなければ、あらかじめディレクトリを移動しておいてください。

対象とするドライブが起動ドライブと異なる場合には、ドライブ名も指定します。

```
A:¥>OPTIMIZE A:
```

このプログラムは、3回リセットが繰り返されるだけで、自動的にCONFIG.SYSとAUTOEXEC.BATが書き換えられます。元のファイルは、CONFIG.OPT、AUTOEXEC.OPTとして保存されます。

実際に使用してみましたが、細かな指定ができないため、日本語FEPの組み込みなどがうまくゆかないことがあるようです（WX IIIの組み込みは失敗しました）。組み込みに失敗した部分があれば、最適化後に手作業で変更をするようにしましょう。

最適化が終了すると、次のようなCONFIG.SYSやAUTOEXEC.BATになっているはずです。

●最適化後のCONFIG.SYS

```
DEVICE=A:¥MEL4WIN¥MELEMM.386 /M D0,D4,E8,EC,F0 /HM
BUFFERS=6
FILES=30
DOS=HIGH,UMB
LASTDRIVE=F
SHELL=A:¥COMMAND.COM A:¥ /P
DEVICEHIGH /L:1 A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥WX3¥WXK.SYS /A5
DEVICE=A:¥WX3¥WX3.SYS /INI=A:¥WX3¥WX3SYS.INI /A5
DEVICEHIGH /L:2 A:¥MEL4WIN¥EXDISK.EXE 6144
```

●最適化後のAUTOEXEC.BAT

```
@ECHO OFF
PATH A:¥WINDOWS;A:¥DOS;A:¥;A:¥MEL4WIN
SET TEMP=E:¥
SET DOSDIR=A:¥DOS
A:¥MEL4WIN¥HYPERDSK C:2048 CW:1024 S
LH DOSKEY /INSERT
PROMPT $P$G
```

OPTIMIZE.COMにより、DOS6に付属のMemMakerと同じような変更が加えられています。

実際には、WXK.SYSとWX3.SYSのところがDEVICEHIGHコマンドになっていましたが、それではFEPの組み込みに失敗してしまうため、DEVICEコマンドに書き換えてあります。

最適化の結果からも、どうもDOS6に付属のドライバ類を用いるほうがよさそうに思えます。

ディスクキャッシュの容量を変更する

『MELWARE for Windows』では、起動後にディスクキャッシュとRAMディスクの容量を変更することが可能です。

HYPERDSK.EXEの割り当て容量を変更するには、組み込み時の指定と同様に実行してください。ただし、最初に確保した容量よりも大きな容量には変更できません。

◉ HYPERDS.EXEの割り当て容量変更オプション

・C:n	：通常時とWindows起動時のキャッシュサイズをnに変更
・C:n CW:m	：通常時のキャッシュサイズをnに、Windows起動時のキャッシュサイズをmに変更

たとえば、通常時の容量を2048Kバイトに、Windows起動時の容量を1024Kバイトに変更するなら、次のように実行します。

```
A:¥>HYPERDSK C:2048 CW:1024
```

RAMディスクの容量を変更する

EXDISK.EXEの割り当て容量を変更するには、Mオプションのあとに変更後の容量を指定します。EMSメモリまたはXMSメモリに余裕があれば、最初の容量よりも大きくすることも可能です。

容量の変更を指定すると、RAMディスク内のファイルはすべて消去されてしまいますから、保存したいファイルがあれば、容量を変更する前にハードディスクなどに保存しておくようにしてください。

たとえば、容量を3072Kバイトに変更するなら、次のように実行します。

```
A:¥>EXDISK M 3072
```

RAM ディスクをコマンドとして組み込む

通常、RAM ディスクは DEVICE コマンドで組み込みますが、『MELWARE for Windows』では、常駐プログラムとして組み込むことが可能です。その場合には、常駐を解除して RAM ディスクのメモリを解放することができます。

たとえば、3072K バイトの RAM ディスクを作成するなら、AUTOEXEC.BAT 内またはコマンドラインから次のように実行します。

```
EXDISK DEVICE=3072
```

同じことを繰り返して、複数の RAM ドライブを作成することもできます（LASTDRIVE の設定の範囲内)。

常駐を解除するには、R オプションを付けて実行してください。

◉ EXDISK の常駐解除

```
A:¥>EXDISK R
```

複数のドライブを作成してある場合には、最後に組み込んだものから順に解放されます。

Step 52

VMAP でメモリの情報を得る

●ポイント●

・『VMAP』を使うと、メモリの様子がわかりやすく表示される。
・ディスクバッファが HMA メモリに収まる範囲を調べたり、UMB メモリの最適化を行う場合には特に便利。

●解　説●

VMAP の特徴とオプション

『VMAP』（VMAP.COM）は、兵藤嘉彦氏（c.mos）によるフリーソフトウェアで、DOS 環境で動作する『Vz エディタ』（ビレッジセンター：作者は同じ兵藤氏）にも付属になっています。

『VMAP』は、メモリなどに関する詳しい情報を表示してくれます。メモリ情報の表示は、DOS6 に付属の MEM コマンドでもできますが、『VMAP』のほうがずっと見やすいと思います。UMB メモリの最適化を行う上で、たいへん役立ってくれるはずです。

『VMAP』には、次の表示オプションがあります。

◉ VMAP の表示オプション

c	コンベンショナルメモリの情報を表示する
u	UMB メモリの情報を表示する
m	コンベンショナルメモリと UMB メモリの情報を表示する（cu と同等）
e	EMS メモリと XMS メモリの情報を表示する
n	複数ブロックをまとめずに表示する

オプションには、「-」や「/」を付ける必要はありません。

オプションを省略すると、UMB メモリ、コンベンショナルメモリ、EMS メモリと XMS メモリの情報が順番に表示されます。

VMAP の表示内容

『VMAP』をオプションを付けずに実行すると、次ページ図のような情報が表示されます（環境や設定によって内容は変わります）。

◉ VMAP の実行例

```
VMAP Version 2.01  Copyright (C) 1989-91 by c.mos

addr PSP  blks   size  owner/parameters           hooked vectors
---- ---- ---- ------  -------------------------  ------------------------------
A666-A7FE   1    6512  <free>
D002 <--    1   13424  wx3k
D34A <--    1    8432  wx3                        29
D55B sys    1    5216  csp
D6A2-D7FE   1    5568  <free>
E903 sys    1    4176  kkcfunc
EA09 sys    1    1488  ramdisk
EA67 sys    1   26800  ctsb16j
F0F3 sys    1   10560  ctmmsys
F388-F400   1    1920  <free>
                       --- UMB total:  83 KB ---
0586 sys    1    2480  himem
0622 sys    1    3408  emm386                     67
06F8 sys    1     224  wxk
0707 sys    1     240  wx3 CON
0717 sys    1    6608  neccd
08B5 sys    4    7824  <config>
0AA2 <--    3    6144  command                    22 23 24 2E
0C25        1     304  <free>
0C39 <--    1   15696  mscdex /D:CD_101 /E
100F <--    1   36304  smartdrv 5120 3072 /Q      09 0A 18 1B 1F 21 25 26 28 DC
18ED <--    1    6320  doskey /insert             07 2F
1A79-9FFF   1  546912  <free>

----- EMS ver4.0 (frame: C000h) -----        ----- XMS ver3.00 -----
handle pages   size  name                    HMA used: 63 KB by DOS
------ -----  ------ --------                EMB free: 8576 KB (max: 11648 KB)
     1     8   128k  WX3K
     2     2    32k  WX3
     3     1    16k
     4     1    16k  FD_HIST$
  free   730 11680k
 total  1468 23488k
```

表示される内容については、次のとおりです。

UMB メモリ、コンベンショナルメモリ

① addr

メモリを占有しているプログラムの開始アドレス（セグメントアドレス）。MEM コマンドの表示より「1」だけ大きく表示される（アドレスの算出方法の違いによる）。

② PSP

PSPのあとにあるプログラムの開始セグメントアドレス。
それぞれ、次の意味がある。

<--	：addrと同じアドレスになっている
sys	：プログラム0008hのシステムコード、データ
空き	：フリーエリア
-????	：フリーエリアの終了アドレス

③ blks

同じプログラムまたはサイズの、連続するメモリブロック数。

④ size

プログラムが使用しているメモリサイズ。

⑤ owner

プログラムの名前。主にプログラム（コマンド）名だが、ほかにも次のものがある。

<free>	：フリー（空き）メモリ
<config>	：CONFIG.SYSで確保したメモリ（BUFFERS、FILESなど）
command	：COMMAND.COM
(env)	：プログラムの環境領域

⑥ parameters

実行されたときのパラメータ（CONデバイスドライバの場合はデバイス名）。

⑦ hooked vectors

プログラムが使っている割り込みベクタ番号。
ただし、<...>タイプのプログラムの場合は表示しない。

⑧ UMB total

UMBに割り当てられている全メモリ。

EMSメモリ

① frame

EMSのページフレームの開始アドレス。

② handle

EMSを利用しているプログラムのハンドル番号。
「free」は、残り全部のEMSメモリ、「total」は全EMSメモリを示す。

③ pages

プログラムの使用メモリ（ページ数）。

④ size

プログラムの使用メモリ（Kバイト数）。

⑤ name

プログラムの名前。

XMSメモリ

① HMA used

HMAの使用メモリと使用プログラム名。

② EMB free

残りEMBメモリ（XMSに割り当てられたメモリ）。
最大ブロックサイズが異なる場合は、「(max: ?KB)」と表示される。

『VMAP』で表示されるなかで、PSPやblksは本書では気にしないでもよい情報になります。

プログラムがメモリに読み込まれるとメモリブロックを形成します。PSPとはProgram Segment Prefixの略で、そのメモリブロックの先頭に作られて、環境文字列（SETコマンドで表示されるデータ）のあるセグメントアドレスや起動時のパラメータなどを情報として持つ領域です。

メモリの最適化を行う上で必要な情報としては、主にSizeとOwner nameになります。

デバイスドライバや常駐プログラムが、どれだけの常駐サイズになっているかということがわかれば、どのUMBメモリの領域に組み込んだらよいかということがおおよそ判断できるからです。

また、UMBメモリのaddrを見れば、デバイスドライバや常駐プログラムがUMBメモリのどのブロックに組み込まれているかということが確認できます。

コンベンショナルメモリに関する情報で、一番下にある<free>の部分のサイズが空き容量になります(この例では546912バイト)。途中に<free>とあっても、連続した領域になっていないため、空き容量としてはカウントできません。

UMBメモリでは、各ブロックの最後にある<free>の部分が空き領域として利用可能です。各ブロックの途中で、プログラムとプログラムの間にある<free>の部分は、コンベンショナルメモリの場合と同様に、連続した領域になっていないため利用できないことになります。

n オプションを付けた場合の表示

『VMAP』をnオプションを付けて表示すると、次のような情報が追加されます（ここではcnオプションを指定)。

◉ cnオプションを付けた場合の実行例

```
VMAP Version 2.01  Copyright (C) 1989-91 by c.mos

addr PSP  blks   size  owner/parameters          hooked vectors
---- ---- ---- ------- ------------------------- -----------------------------
0586 sys    1    2480  himem
0622 sys    1    3408  emm386                    67
06F8 sys    1     224  wxk
0707 sys    1     240  wx3 CON
0717 sys    1    6608  neccd
08B5 sys    1    4432  <files>
09CB sys    1      80  <fcbs>
09D1 sys    1    1024  <buffers>
0A12 sys    1    2288  <lastdrv>
0AA2 <--    1    5056  command cn                22 23 24 2E
0BDF        1      64  <free>
0BE4 0AA2   1    1024  command (env)
0C25        1     304  <free>
0C39 <--    1   15696  mscdex /D:CD_101 /E
100F <--    1   36304  smartdrv 5120 3072 /Q     09 0A 18 1B 1F 21 25 26 28 DC
18ED <--    1    6320  doskey /insert            07 2F
1A79-9FFF   1  546912  <free>
```

owner の部分で、通常は<config>としてまとめられている情報が、次のように展開されて表示されます。

<buffers>	：BUFFERS で確保されたディスクバッファ
<files>	：FILES で確保されたメモリ
<fcbs>	：FCBS で確保されたメモリ
<lastdrv>	：LASTDRIVE で確保されたメモリ

また、COMMAND.COM も環境部分が (env) として分けて表示されています。

BUFFERS で確保したディスクバッファが、HMA メモリからはみ出た場合には、<buffers>の値が極端に大きくなります（収まっていれば 1K バイト程度）。

Step 53

SETPATHで長いパスを設定する

●ポイント●

・『SETPATH』を利用すると、約300バイトまでの検索パスが設定できる。
・『SETPATH』は、PATHコマンドの代わりとして利用できるほか、あとから検索パスを追加することもできる。
・『SETPATH』によって設定した128バイトを超えるパスは、『SETPATH』で確認することができる。

●解 説●

SETPATHの特徴と使い方

『SETPATH』(SETPATH.EXE)は、Kenny氏（BOSTON倶楽部）によるフリーソフトウェアで、128バイトを超える長い検索パスが設定できるようになります。これなら、ディレクトリが多くなっても、検索パスが足らなくて困ることは少なくなるはずです。

『SETPATH』は、AUTOEXEC.BAT内でPATHコマンドの代わりに用いることができます。

```
A:¥TOOL¥SETPATH A:¥;A:¥WINDOWS;A:¥DOS;A:¥BATCH;A:¥TOOL;……
```

検索パスを追加する場合は、追加したいパスだけを指定してください。

●検索パスを追加する

```
A:¥>SETPATH B:¥GAME1;B:¥GAME2
```

追加するパスの最後には、セミコロン(;)があってもなくてもかまいません。

すでに登録されているパスを設定すると二重に設定されますが、実行上の問題はないはずです。

全部で、およそ300バイト分の検索パスが設定できます。ただし、128バイトを超えた分はSETコマンドで確認することができません。その場合には、SETPATHコマンドをオプションなしで実行してください。

● SETPATHによるパスの確認

```
A:¥>SETPATH
PATH=A:¥;A:¥WINDOWS;A:¥DOS;A:¥JSLIB;A:¥BATCH;A:¥TOOL;A:¥MYT;A:¥MT
;A:¥NORTON;A:¥MDEV¥IOSPRO;A:¥MEL4WIN;A:¥EDITOR¥VZ;A:¥EDITOR¥MIFES
;A:¥ATOK8;B:¥GAME1;B:¥GAME2;
```

小文字で指定しても、登録時には大文字に変換されます。

不必要に検索パスを長くすると、あとのほうのディレクトリの検索が遅くなってしまいますから、頻繁に利用するパスははじめのほうに設定してください。

なお、付属のMCB.EXEは、メモリのつながりを表示するプログラムです。SETPATH.EXEを使用していて、異常が発生した場合に、作者に状況を知らせるために実行します。データを送る際は、パソコン通信の電子メールを使用してください。

Step 54

ENV_DRV でドライブ情報を環境変数に登録する

●ポイント●

- 『ENV_DRV』は、ドライブの種類などの情報を環境変数に登録する。
- 『ENV_DRV』を用いると、ドライブ構成に変更があっても、作業ドライブを RAM ディスクなどに正しく設定することができる。

●解 説●

ENV_DRV の特徴とオプション

PC-9800 シリーズの仕様では、システムを起動するドライブによって、個々のディスク装置に割り当てられるドライブ名が変わってきてしまいます。しかし、作業用ドライブとして RAM ディスクを指定しているような場合には、ドライブ名が変わるとバッチファイルが誤動作するなど、困ってしまうことになります。

『ENV_DRV』(ENV_DRV.EXE)は、山下 良輔氏 (ryama) によるフリーソフトウェアで、ドライブの種類などの情報を環境変数に登録してくれます。AUTOEXEC.BAT で『ENV_DRV』を実行しておけば、ドライブ構成の変更にも対応したバッチファイルを利用できてたいへん便利です。

『ENV_DRV』には、次のオプションがあります。

● ENV_DRV のオプション

-f	フロッピーディスクの設定(環境変数 FD?にセットされる)
-h	ハードディスクの設定(環境変数 HD?にセットされる)
-o	MO ディスクの設定(環境変数 OD?にセットされる)
-r	RAM ディスクの設定(環境変数 RD?にセットされる)
-n	リモートドライブの設定(環境変数 ND?にセットされる)
-c	CD-ROM ドライブの設定(環境変数 CD?にセットされる)
-s	SUBST で割り当てられた仮想ドライブの設定(環境変数 SD?にセットされる)
-l	リモートリンクディスクの設定(MAXLINK/LAPLINK:環境変数 LD?にセットされる)
-x	圧縮ドライブを識別する(DiskXII/Stacker/DoubleSpace:環境変数 XD?にセットされる)
-e	メディア交換可能なドライブをフロッピーディスクとみなす
-u	ダミーの RAM ディスクは無視する
-2	仮想およびリモートドライブは無視する(MS-DOS Ver2.xx とみなす)
-:	ドライブ名に「:」のない形式で設定
-p	CPU/FPU の種類を設定
-k	機種・モードを設定する
-t	ルートの COMMAND.COM の環境を設定する
-i	環境変数の設定を一切しない
-v	環境変数に設定すると同時に画面にも識別結果を表示する
-$	機種・モードにかかわらず画面表示を英語にする
+＜DeviceName＞=?D	CONFIG.SYS にドライバを組み込む必要のあるブロックデバイスで、RAM ディスク以外のドライブに設定したいとき、そのデバイス名と種類を指定する

f、h、o、r、n、c、s のいずれも指定しなければ、-fhorncs が指定されたものとして実行されます。

ENV_DRV を利用する場合には、CONFIG.SYS の SHELL コマンドで環境領域を広げておくようにするほうがよいでしょう。512 程度の設定で足りるはずです。

とりあえず、コマンドラインから次のことを試してみてください。

```
A:¥>ENV_DRV
A:¥>SET
```

環境変数にセットされた内容が確認できるはずです。

ページの都合により、これらのすべてについて解説することはできませんから、主に用いると思われる RAM ディスクに関する設定方法について解説します。その他のオプションについては、『ENV_DRV』のドキュメントファイルを参照してください。

常に RAM ディスクを作業用ドライブにする

ドライブ構成が変更になってしまうことがあるシステムで、常に RAM ディスクを作業用ドライブとして割り当てるなら、AUTOEXEC.BAT で次のように設定するとよいでしょう。

◉ RAMDISK のドライブ名を環境変数 TEMP に設定する

```
ENV_DRV -hr
SET TEMP=%RD1%¥
IF "%RD1%"=="" SET TEMP=%HD1%¥
IF "%HD1%"=="" SET TEMP=.
```

まず、ENV _DRV を-hr スイッチ付きで実行して、ハードディスクと RAM ディスクのドライブ名を環境変数にセットしておきます。

次に、環境変数 TEMP に RAM ディスクのドライブが入っている環境変数 RD1 を代入して、そのあとに「¥」を加えておきます。たとえば、RAM ディスクが D ドライブなら、環境変数 TEMP は「D:¥」になるわけです。

万一、環境変数 RD1 が空……つまり、RAM ディスクが作成されていなければ、環境変数 TEMP にはハードディスクのドライブを割り当てます。

同様に、ハードディスクもなければ、TEMP にはカレントディレクトリが割り当てられます（「.」はカレントディレクトリを示す記号です)。

Step 55

HSBで高速再起動を実現する

●ポイント●

- 『HSB』を使うと高速に再起動が行える。
- 環境ファイルを用意しておくと、『HSB』で環境を使い分けることができる。
- 『HSB』では、環境変数HSBEXTとHSBDIRを設定しておく。
- 『HSB』を常駐させてホットキーの機能を使う場合には、書き込みキャッシュに注意。

●解 説●

HSBの特徴

『HSB.EXE』は、森河正男氏（Masao）によるフリーソフトウェアで、システムの再起動を高速で行えるほか、CONFIG.SYSやAUTOEXEC.BATの内容を入れ替えたり、別ドライブから再起動する機能があります。本書の執筆にあたっては、この『HSB』にずいぶんお世話になりました。

『HSB』には、実に多くのオプションがあります。本書では、それらについて解説することはできませんが、付属のドキュメントでたいへん詳しく解説されていますから、ぜひそちらをお読みいただきたいと思います（プリンタで出力することをお勧めします）。

HSBの基本的な使い方

ここでご紹介するのは、たくさんある使用法のほんの一例にすぎません。

まず、CONFIG.SYSの先頭に、次の1行を加えてください。「A:¥TOOL¥」の部分は『HSB』をインストールしてあるディレクトリに合わせます。

◉ HSBを設定するためにCONFIG.SYSに組み込む

```
DEVICE=A:¥TOOL¥HSB¥EXE VC Y-
```

これで、そのまま再起動してください。HSBがあなたの環境に合わせて変更されます。特に動作環境を変更する必要がない、というのなら、もうこの1行を削除してもかまいませんが、HSBはメモリに常駐するわけではありませんから、加えたままにしておいてよいでしょう。システム環境が変わってしまうと再設定が必要になります。

VCオプションは、起動時に環境検査を行って、もしも現在のHSB.EXEがもっている環境情報と異なっていれば、書き換えを行う指定です。

Y-オプションは、HSB.EXEをメモリに常駐させないという指定です。

この手順を済ませておけば、次回から次のように実行するだけで、高速に再起動が行われるようになります。試してみてください。

◉ HSBによる高速再起動

```
A:¥>HSB
```

環境ファイルを使い分ける

『HSB』には、CONFIG.SYS と AUTOEXEC.BAT（これらを合わせて「環境ファイル」と呼ぶことにします）の内容をほかのファイルと入れ替えたかたちで再起動する機能があります。実際にファイル名の変更が行われるわけではないので、万一、起動に失敗しても、通常通りに再起動を行えばよいのです。

●環境ファイル切り換え起動の手順

まず、複数の環境ファイルの内容を拡張子だけを変えて作成してください。たとえば、Windows 用ならば、CONFIG.WIN と AUTOEXEC.WIN というようにしておきましょう。

拡張子を変更しただけのファイルは、1つのディレクトリにまとめておきます。ここでは、CONAUT というディレクトリ名にします。

次に、AUTOEXEC.BAT で、次の環境変数がセットされるようにしてください。

◉再起動用の環境変数の設定

```
SET HSBEXT=TMP
SET HSBDIR=CONAUT
```

環境変数 TMP は、再起動の際に用いられる仮の環境ファイルの拡張子になります。特に指定はありませんから、ここでは「TMP」としておきます。

環境変数 HSBDIR は、作成した環境ファイルの複製を入れてあるディレクトリの名前です。ここには、ドライブ名は指定しないでください。「A:¥DOS¥CONAUT」であれば、「DOS¥CONAUT」と指定します。

こうしておいて、一度再起動を行いましょう。

HSB で環境ファイルを指定するには、次のようにします。

```
A:¥>HSB :WIN
```

「:」は、カレントドライブから再起動するという指定で、「WIN」は環境変数 HSBDIR で指定したディレクトリ内の、対応する拡張子の環境ファイルを用いるという指定になります。

HSBの実行例

いくつかのパターンをお見せします。

◉オプションなしで実行

```
A:¥>HSB
```

オプションを何も指定しなければ、前回の起動ドライブから再起動が行われます。環境ファイルは、起動ドライブのルートディレクトリにあるCONFIG.SYSとAUTOEXEC.BATが使われます。

◉カレントドライブから起動

```
A:¥>HSB :
```

コロン(:)を付けると、カレントドライブから再起動が行われます。

環境ファイルは、カレントドライブのルートディレクトリにあるCONFIG.SYSとAUTOEXEC.BATが使われます。

◉ドライブを指定して再起動

```
A:¥>HSB B:
```

Bドライブから再起動を行います。

環境ファイルは、Bドライブのルートディレクトリにある CONFIG.SYS と AUTOEXEC.BAT が使われます。

ホットキーを設定する

『HSB』は、コマンドとして実行するほか、メモリに常駐させておくと、特定のキー操作（ホットキー）だけで再起動が行えるようになります。

ただし、ホットキーを利用して再起動を行う場合、SMARTDRV.EXEのような書き込みキャッシュを行うディスクキャッシュドライバを使っていると、うっかりキャッシュ内のデータを書き込まずに再起動を行ってしまう危険があります（コマンドとして実行する際には、キャッシュ内容が強制的に書き込まれるようになっているので安全です）。

自分は特にそそっかしいと思ったら、ホットキーを利用する場合には書き込みキャッシュをオフにしておくほうがよいかもしれません。

メモリに常駐させるには、CONFIG.SYSの記述を次のように変更してください。

◉ HSB.EXEをメモリに常駐させる設定（CONFIG.SYS）

```
DEVICE=A:¥TOOL¥HSB¥EXE VC T2 I26 Y2
```

VCは、すでに解説した環境検査を行うオプションです。

T2は、ホットキーを行う指定になります。

I26は、再起動時にFM音源の初期化を行う指定です。

Y2は、HSB.EXEを常駐させるための指定です。

この設定で起動して、CTRL＋GRPH＋DELキーを押してください。直ちに再起動が行われるはずです。

もしも、何か不具合が発生した場合には、ほかのプログラムとの相性が悪いということが考えられます。対処方法は、『HSB』のドキュメントの12ページ以降に詳しく書かれていますから、そちらを参照してください。

Step 56

HSBMで再起動メニューを作る

●ポイント●

- 『HSBM』は、『HSB』に対応したメニューソフト。
- 『HSBM』を使う前に、環境変数HSBMをセットしておく。
- 『HSBM』では、INSキーを押してメニューに登録する。
- 『HSBM』では、登録時にエディタを起動して環境ファイルやバッチファイルの作成が可能。

●解説●

HSBMの特徴と準備作業

『HSBM.EXE』は、KEi氏によるフリーソフトウェアで、Masao氏作の『HSB』による再起動がメニューから行えるほか、バッチファイルによるFEPの切り替えや、アプリケーションの起動なども行うことができます。

『HSBM』を起動する前に、AUTOEXEC.BATで環境変数「HSBM」を設定しておきましょう。ここには、『HSBM』のファイルがあるディレクトリを指定しておきます。

◉環境変数HSBMを設定する

```
SET HSBM=A:¥TOOL
```

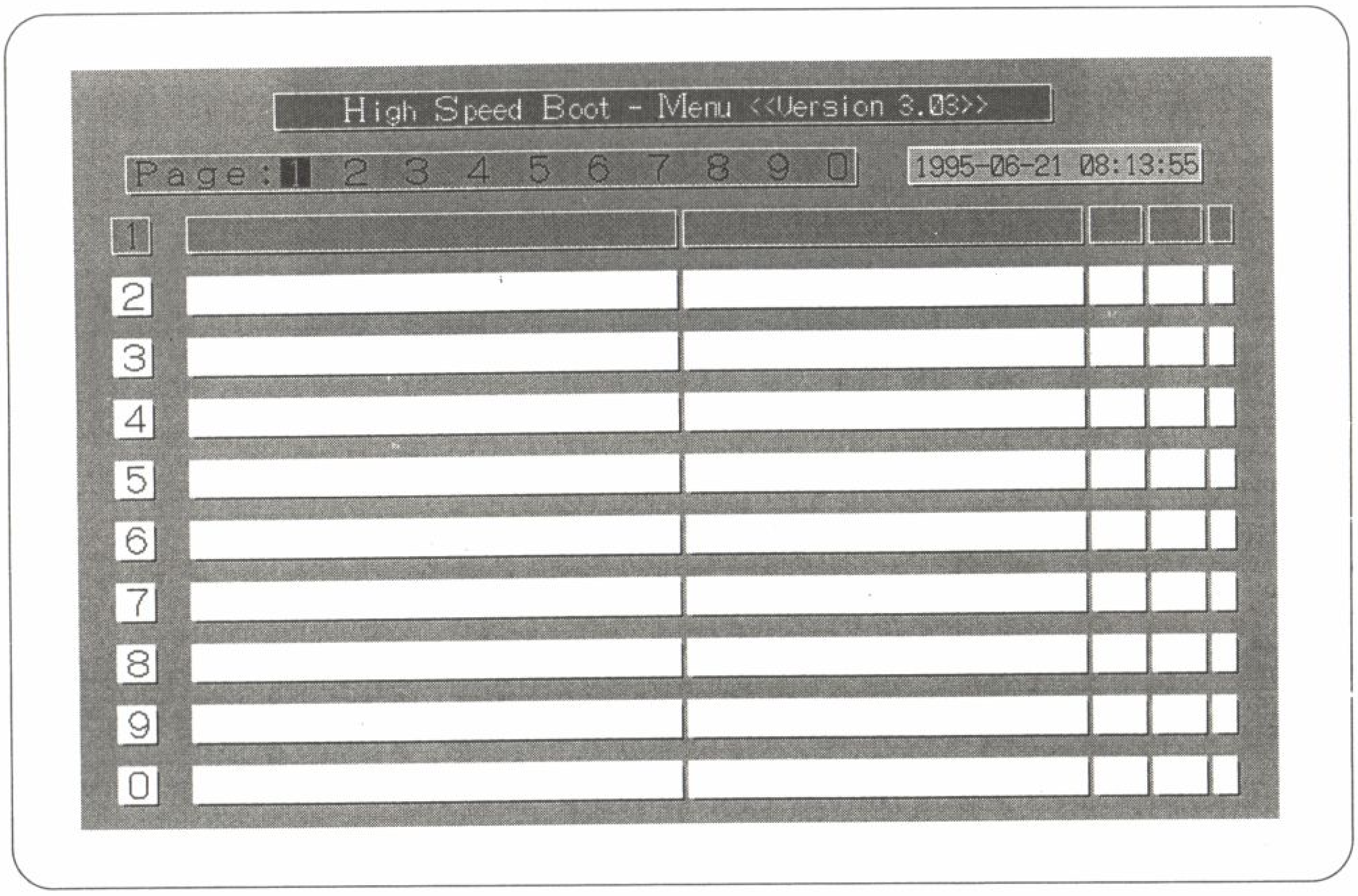
環境変数「HSBM」で指定したディレクトリには、『HSBM』で必要な環境ファイルが作成されます。

環境変数の設定が終わったら、再起動するか、またはコマンドラインから環境変数「HSBM」をセットしておいて、『HSBM』を起動してください。

◉ HSBMを起動する

```
A:¥>HSBM
```

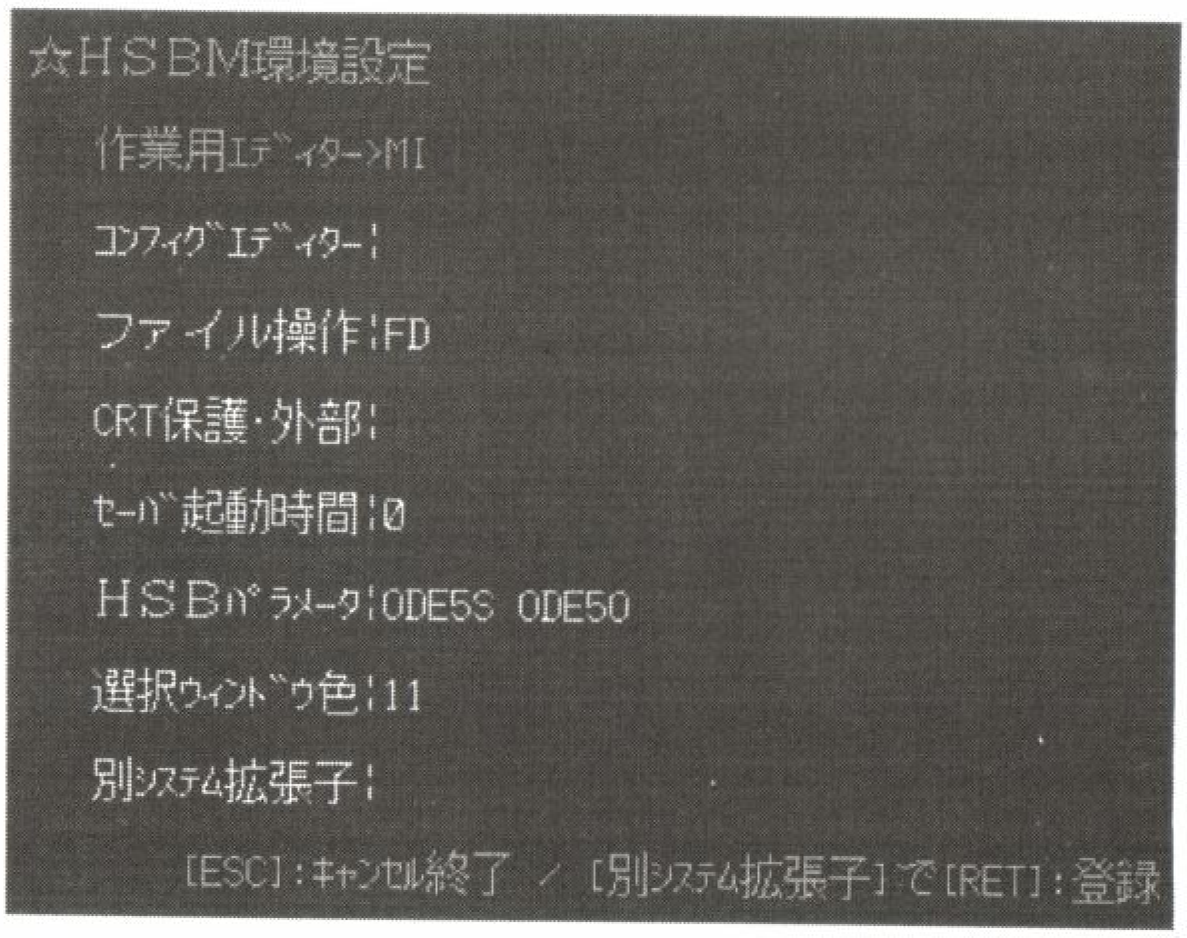
◉ HSBMのメニュー画面



起動時に表示されるメニュー画面には、まだ何も登録されていません。

最初に、『HSBM』の環境設定を行うため、TAB キーを押してください。

◉ HSBM の環境設定画面



ここでは、以下の項目について設定します。

●作業用エディター

環境ファイルやバッチファイルを作成するためのエディタを登録します。エディタを起動するバッチファイル名でもかまいません。

たとえば、『Vz エディタ』なら「VZ」、『MIFES』なら「MI」または「MIFES」のように入力します。もし、それらのエディタがなければ、DOS6 に付属の「SEDIT」でもかまいませんが、複数ファイルの同時編集が可能なエディタを使用することをお勧めします。

●コンフィグエディター

コンフィグ（CONFIG）ファイルを作成するためのエディタを登録します。省略すると、作業用エディタと同じになります。

いずれ、作者によってコンフィグ編集用の専用エディタが提供される予定とのことですが、現在のところは省略しておいてよいでしょう。

●ファイル操作

ファイル操作を行うコマンドを登録します。

たとえば、『エコロジー』なら「EC」、『FD』（出射厚氏作のフリーソフトウェア）なら「FD」とします。ファイル操作を行わなければ、省略してかまいません。

● CRT 保護・外部

外部スクリーンセーバを指定します。指定しなければ、内部スクリーンセーバが起動されます。

スクリーンセーバとは、画面の焼き付きを防止するため、一定時間入力がなければ起動する画面保護プログラムのことです（Windows 環境ではおなじみですね）。

●セーバ起動時間

スクリーンセーバが起動する時間を分単位で指定します。0 を指定すると、自動起動されなくなります。通常、3〜5 分くらいがよいと思います。

なお、スクリーンセーバとは相性の悪いドライバや常駐プログラムもあるため、何か不具合が発生したら、スクリーンセーバを使わない設定にしてみてください。

● HSB パラメータ

HSB による再起動を行う際に、HSB に渡すパラメータ（オプション）を指定します。通常、ここは変更の必要はないと思います。

●選択ウィンドウ色

メニューで選択されている項目の背景色を指定します。色は、次の番号から指定してください。1〜6 までは明るい色、それ以降は落ち着いた感じの色になります。

1:青	2:緑	3:水色	4:赤	5:紫	6:黄色	7:白	8:ダークグレー	9:紺
10:緑(暗)	11:水色(暗)	12:赤(暗)	13:グレー	14:黄色(暗)				

●別システム拡張子

同じドライブから別バージョンのシステムを起動する場合、システムファイルに付ける拡張子を指定します。別システムを起動しなければ、指定する必要はありません。

各項目の設定が終わったら、一番下の「別システム拡張子」の項目のところで、⏎キーを押してください。設定内容が保存されます。

メニューへの登録方法

『HSBM』のメニューに登録するには、↑ ↓キーで登録したいメニューに移動しておいてから、INSキーを押してください。メニューは、1〜0までの10ページに、それぞれ10項目が登録可能です。

ここでは、次の項目について入力します。

●ソフト名

起動するアプリケーションの名称や、再起動の目的などを入力します。

●ソフト種

起動するアプリケーションの種類や、再起動の内容などについてのコメントなどを入力します。自分（またはほかの人）がわかりやすいと思う内容を適当に入力してください。

●起動種

実行する内容について、← →キーを使って次のなかから選択します。

- **HSB起動・ノーマル**：HSBによる再起動を行う。
- **HSB起動・別システム**：HSBによる別システムを用いた再起動を行う。
- **バッチファイル起動**：AUTOEXEC.〜だけを実行する。
- **ダミー・タイトル(非起動)**：何も起動しない(コメントなどを入力する)。

●ディレクトリー

CONFIG.～、AUTOEXEC.～があるパス名を指定します。

●拡張子

実行するCONFIG.～やAUTOEXEC.～の拡張子を指定します。

●ショート・カット

アルファベット1文字を指定します。そのアルファベットキーを SHIFT キーと一緒に押すと、直ちにメニューの内容が実行されます。ほかのメニューと重複しないアルファベットを指定してください。

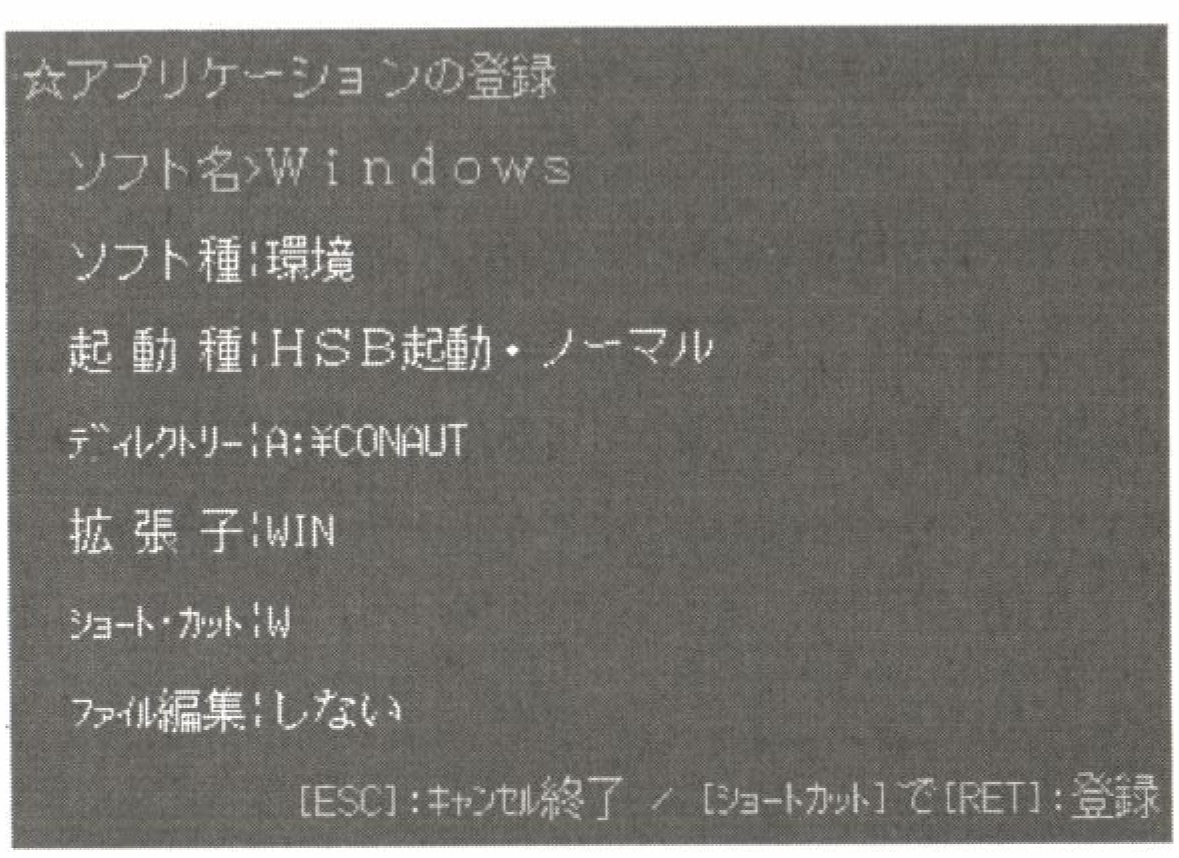
●ファイル編集

CONFIG.～、AUTOEXEC.～を編集するかどうかを指定します。← →キーを使って「する」「しない」を選択してください。ただし、ファイルが存在しない拡張子を指定してあると、「する」以外は選択できません。

これらの項目のうち、省略ができないのは、「ソフト名」「起動種」「ディレクトリー」「拡張子」です。ただし、起動種によっては、「ディレクトリー」と「拡張子」が省略可能になります。

たとえば、『HSB』を用いて通常の再起動を行いたい場合には、次のように入力します。

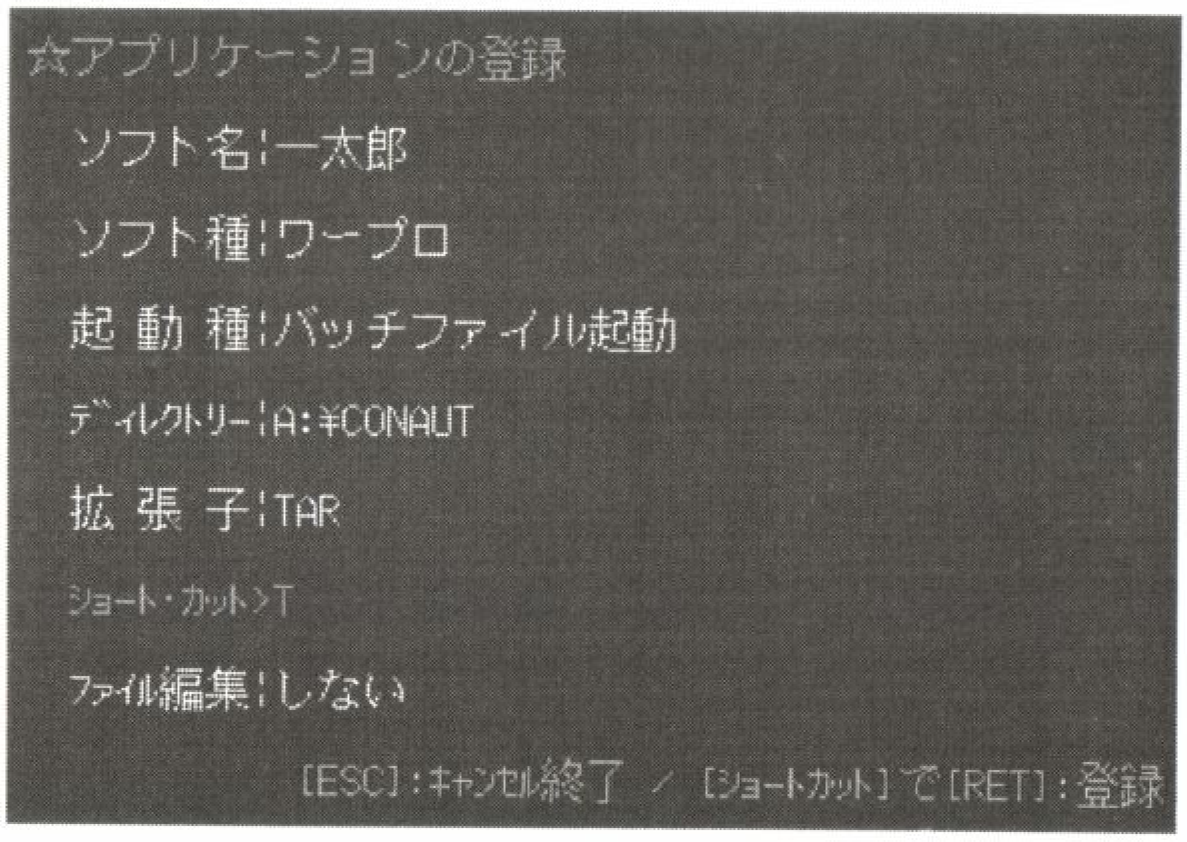
◉ HSBによる再起動を行う場合の設定例



ここでのポイントは、まず「起動種」に「HSB起動・ノーマル」を指定すること。それから、「ディレクトリー」にCONFIG.～、AUTOEXEC.～のあるパス名を入力して、「拡張子」にそれぞれの拡張子を指定することです。

また、再起動を行わずに、アプリケーションを起動するには、次のように入力します。

◉アプリケーション起動を行う場合の設定例



この場合には、「起動種」に「バッチファイル起動」を指定すること。それから、「ディレクトリー」にAUTOEXEC.～のあるパス名を入力して、「拡張子」に、その拡張子を指定することです。

このあと、登録時にファイル編集を「する」にしておき、アプリケーション起動用に使っているバッチファイルを「AUTOEXEC.～」という名前に変更して、指定したディレクトリに作成しておいてください。

たとえば、TARO.BATを一太郎を起動するために使っている場合、AUTOEXEC.TARのような名前に変えて使うのです。『HSBM』を起動する前に、あらかじめ、この作業を行っておいてもかまいません。

アプリケーション起動を行うと、メモリ内に『HSBM』が常駐して、アプリケーションが終了すると、再び『HSBM』のメニューに戻るようになります。

メニュー項目の削除

メニューに設定した内容を再編集するには、該当する項目のところで INS キーを押してください。設定画面が表示されます。

また、設定済みの項目を削除するには、該当する項目のところで DEL キーを押します。ただし、指定されていた CONFIG.〜や AUTOEXEC.〜も同時に削除されますから注意してください。

デバイスドライバの設定

たとえば、通常は Windows 環境を中心に利用していて、『HSBM』を用いて DOS のアプリケーションを起動するような場合、FEP を組み込むために AUTOEXEC.〜のなかで ADDDRV コマンドを実行させてもかまいません。

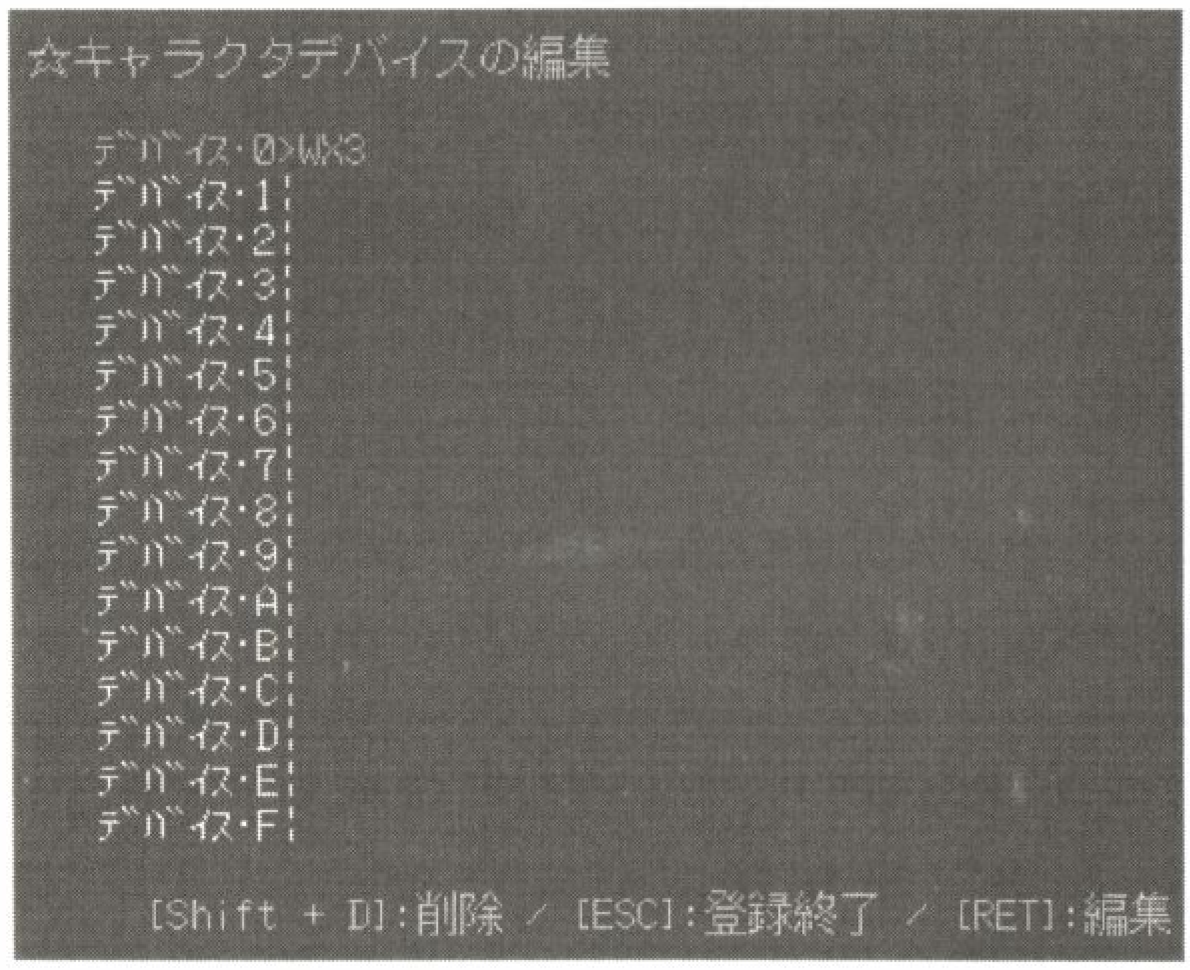
◉ ADDDRV コマンドを用いた AUTOEXEC.〜の例

```
ADDDRV A:¥DRV¥WX3.DEV
MT
DELDRV
```

しかし、FEP のようにしばしば用いるものは、『HSBM』に登録しておくと便利です。

デバイスドライバを登録するには、『HSBM』のメニュー画面で、ROLL UP キーを押してください。

◉デバイスドライバの登録画面



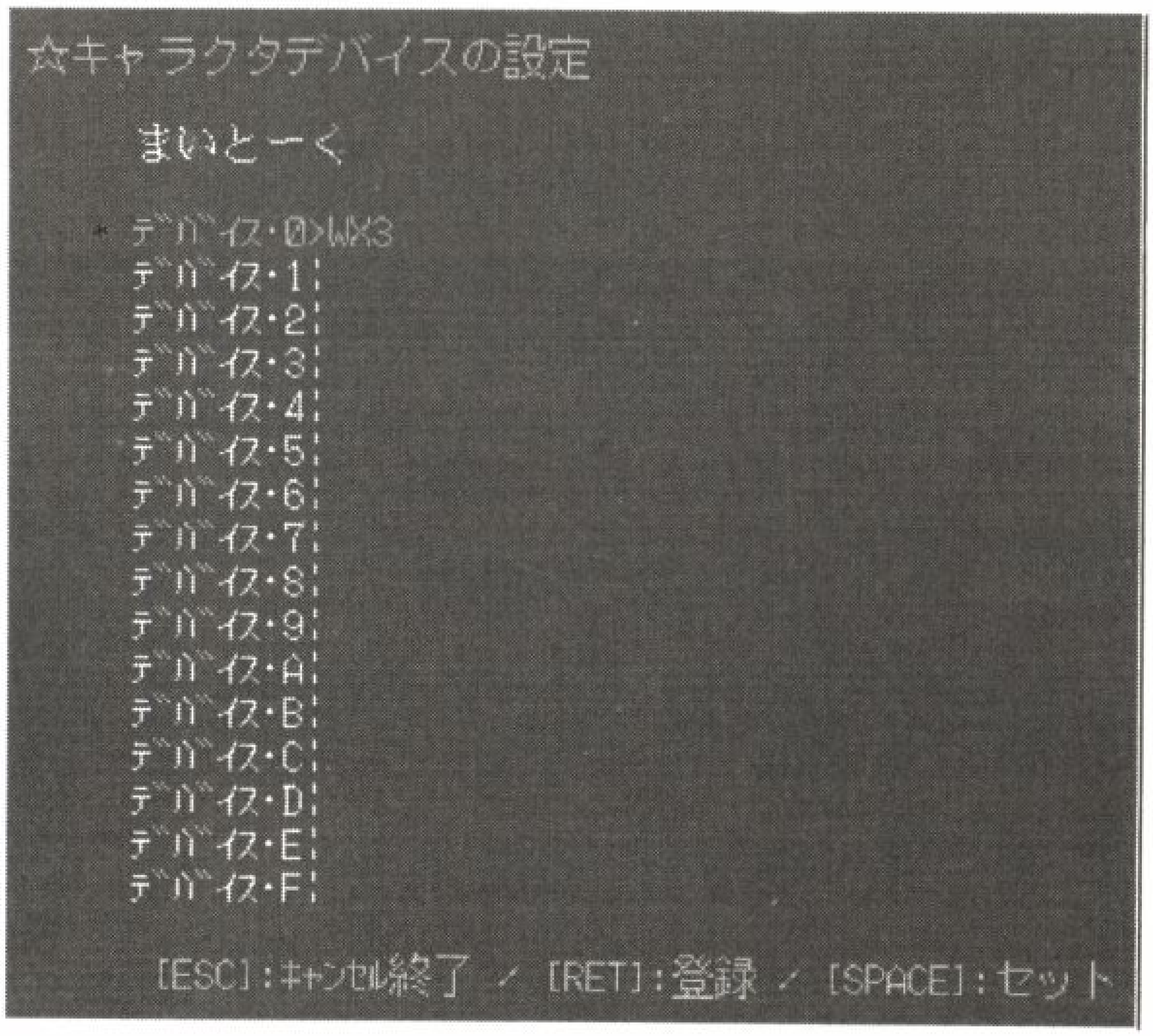
まず、登録したい場所(0〜F)を選んで ⏎ キーを押し、何かコメントを入力してください（組み込むドライバの名前など)。すると、エディタが起動されて、HSBM_DEV.C01 というファイル(01 の部分は変わります)の編集に移りますから、ADDDRV コマンドで用いる設定ファイルの書式で入力してください。たとえば、ATOK8 を組み込むなら、次のようになります(DEVICEHIGH コマンドも利用できます)。

```
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
```

ファイルを保存してエディタを終了すると、登録作業は完了して、元のメニュー画面に戻ります。

こうして登録したデバイスドライバを組み込むには、まずメニューでデバイスドライバを組み込んで使いたいアプリケーションの項目に移動しておいて、ROLL DOWN キーを押してください。

●デバイスドライバの指定画面



登録されているデバイスドライバの一覧が表示されますから、そのなかから組み込みたいもののところでスペースキーを押してください。先頭にマーク（*）が付けられます。複数が設定可能ですが、メニューの上から順に組み込まれることになります。

ESC キーで元のメニューに戻り、設定した項目を実行してみてください。デバイスドライバの組込みが行われてから、アプリケーションが起動されるはずです。アプリケーションを終了すると、元のメニューに戻りますが、そのときにはデバイスドライバは外されています。

なお、設定されたデバイスドライバを削除するには、設定画面で SHIFT ＋ D キーを押してください。その際には、対応する HSBM_DEV.～も削除されます。

その他の機能

これまでに解説した以外にも、『HSBM』にはさまざまな機能があります。それらのすべてを解説できませんので、あとはドキュメントファイルを参照してください。最後に、キー操作方法を一覧にしておきます。

◉ HSBM のキー操作一覧

⏎	メニューの実行（SHIFT ＋アルファベットも同じ）
スペース	メニューにマークをして、順番を入れ替える
TAB	環境設定
INS	メニューへの登録
DEL	メニューの削除
ROLL UP	デバイスドライバの登録
ROLL DOWN	デバイスドライバの組み込み設定
BS	時間指定起動の設定
HOME CLR	ファイル操作コマンドの起動
SHIFT ＋ HOME CLR	作業エディタの起動
SHIFT ＋ TAB	外部コマンドの設定
SHIFT ＋ファンクション (1〜10)	外部コマンドの実行
SHIFT ＋ ESC	DOS 画面へ（EXIT コマンドで戻る）
HELP	ヘルプメッセージの表示
SHIFT ＋ HELP	ショートカットキーの一覧表示
SHIFT ＋ ⏎	CONFIG.〜と AUTOEXEC.〜の表示
／	スクリーンセーバの起動
CTRL ＋ ⏎	フロッピーディスクの一番若い番号のドライブから再起動
ファンクション(1〜10)	ドライブを指定した再起動（f･1 が A ドライブ……）
ESC	HSBM の終了

付属ディスクの使い方

本書付属のフロッピーディスクには、次のプログラムなどが収録されています。

- VMAP　：兵藤嘉彦（c.mos)/フリーソフトウェア
- SETPATH：Kenny（BOSTON 倶楽部)/フリーソフトウェア
- ENV_DRV：山下良輔（ryama)/フリーソフトウェア
- HSB　：森河正男（Masao)/フリーソフトウェア
- HSBM　：KEi/フリーソフトウェア
- CONAUT：CONFIG/AUTOEXEC ファイルのサンプル集

それぞれ、圧縮されない状態でディレクトリに入っていますから、ファイル操作ソフトや COPY コマンドを用いてもインストールが可能です。

●インストール用バッチファイルによるインストール

なお、VMAP/SETPATH/ENV_DRV/HSB/HSBM をインストールするように、専用のバッチファイルを用意してあります。次のように実行してください。

```
A:¥>INST ドライブ名
```

たとえば、A ドライブにインストールするなら、「INST A:」とします。

インストールされるディレクトリは「¥TOOL」です。もしも、同名のディレクトリがあれば、そこへコピーされます。

● README.DOC について

収録されているファイルに関する詳しい解説は、ルートディレクトリにある README.DOC のなかにあります。それぞれのソフトウェアについて、あとからのバージョンアップによって本文中の解説と食い違うところが生じているかもしれませんから、必ずお読みいただきたいと思います。

●環境ファイルのサンプル集について

『HSB』や『HSBM』のための参考資料として、CONFIG.～と AUTOEXEC.～のサンプル集も用意しました。まったく同じ内容で動作するとは限りませんが、ゼロから作成するよりは楽になると思います。環境によって利用してもよいファイルが異なりますから、これについても README.DOC で内容を確認の上ご利用ください。

フリーソフトウェア使用上の注意事項

「フリーソフトウェア」とは、主にネットワーク内に登録されているプログラムのうち、作者が指定した条件のもとで自由な使用が認められているものをいいます。

フリーソフトウェアの著作権は、それぞれの作者にあります。著作権が放棄されているわけではないので、作者の示した条件から外れるような使い方をした場合には、著作権を侵害したことになります。

特に、筆者からお願いしたいのは以下の点です。

・本書では、ネットワークから圧縮状態で受け取ったファイルを、筆者から特にお願いして、非圧縮状態で収録させていただいています。したがって、本書のファイルをそのままほかのメディアで利用したり、ネットワーク内に転載するようなことは絶対にしないでください。

・それぞれのフリーソフトウェアを利用する前に、ドキュメントファイルには必ず目を通すようにしてください。

・各フリーソフトウェアを使用した上で発生した事故などに対して、作者、著者（寺口俊伸）、およびエーアイ出版は一切の責任を負いません。

細かな使用条件などについては、それぞれのドキュメントファイルを参照してください。

●各フリーソフトウェアの登録場所

それぞれのフリーソフトウェアは、平成7年6月20日現在、NIFTY-Serve内の次のライブラリに登録されています。

◉『VMAP』・『SETPATH』・『ENV_DRV』

FGALTL（FGAL・ツール）
3番ライブラリ【デバドラ・環境変数・バッチ】

◉ HSB

FGALTL（FGAL・ツール） 10 番ライブラリ【その他のツール一般】

◉ HSBM

FGALTL（FGAL・ツール） 7 番ライブラリ【メニュー関連ソフト】

これらのライブラリには、最新版が登録されているはずですから、もしも可能ならばライブラリを覗いてみることをお勧めします。

ただし、フォーラム改編によって、これらの登録場所が変更になっていることもあります。

付録◎ 2 進、10 進、16 進対応表

10進	16進	2進	10進	16進	2進
0	00H	0000 0000	31	1FH	0001 1111
1	01H	0000 0001	32	20H	0010 0000
2	02H	0000 0010	33	21H	0010 0001
3	03H	0000 0011	34	22H	0010 0010
4	04H	0000 0100	35	23H	0010 0011
5	05H	0000 0101	36	24H	0010 0100
6	06H	0000 0110	37	25H	0010 0101
7	07H	0000 0111	38	26H	0010 0110
8	08H	0000 1000	39	27H	0010 0111
9	09H	0000 1001	40	28H	0010 1000
10	0AH	0000 1010	41	29H	0010 1001
11	0BH	0000 1011	42	2AH	0010 1010
12	0CH	0000 1100	43	2BH	0010 1011
13	0DH	0000 1101	44	2CH	0010 1100
14	0EH	0000 1110	45	2DH	0010 1101
15	0FH	0000 1111	46	2EH	0010 1110
16	10H	0001 0000	47	2FH	0010 1111
17	11H	0001 0001	48	30H	0011 0000
18	12H	0001 0010	49	31H	0011 0001
19	13H	0001 0011	50	32H	0011 0010
20	14H	0001 0100	51	33H	0011 0011
21	15H	0001 0101	52	34H	0011 0100
22	16H	0001 0110	53	35H	0011 0101
23	17H	0001 0111	54	36H	0011 0110
24	18H	0001 1000	55	37H	0011 0111
25	19H	0001 1001	56	38H	0011 1000
26	1AH	0001 1010	57	39H	0011 1001
27	1BH	0001 1011	58	3AH	0011 1010
28	1CH	0001 1100	59	3BH	0011 1011
29	1DH	0001 1101	60	3CH	0011 1100
30	1EH	0001 1110	61	3DH	0011 1101

10進	16進	2進	10進	16進	2進
62	3EH	0011 1110	96	60H	0110 0000
63	3FH	0011 1111	97	61H	0110 0001
64	40H	0100 0000	98	62H	0110 0010
65	41H	0100 0001	99	63H	0110 0011
66	42H	0100 0010	100	64H	0110 0100
67	43H	0100 0011	101	65H	0110 0101
68	44H	0100 0100	102	66H	0110 0110
69	45H	0100 0101	103	67H	0110 0111
70	46H	0100 0110	104	68H	0110 1000
71	47H	0100 0111	105	69H	0110 1001
72	48H	0100 1000	106	6AH	0110 1010
73	49H	0100 1001	107	6BH	0110 1011
74	4AH	0100 1010	108	6CH	0110 1100
75	4BH	0100 1011	109	6DH	0110 1101
76	4GH	0100 1100	110	6EH	0110 1110
77	4DH	0100 1101	111	6FH	0110 1111
78	4EH	0100 1110	112	70H	0111 0000
79	4FH	0100 1111	113	71H	0111 0001
80	50H	0101 0000	114	72H	0111 0010
81	51H	0101 0001	115	73H	0111 0011
82	52H	0101 0010	116	74H	0111 0100
83	53H	0101 0011	117	75H	0111 0101
84	54H	0101 0100	118	76H	0111 0110
85	55H	0101 0101	119	77H	0111 0111
86	56H	0101 0110	120	78H	0111 1000
87	57H	0101 0111	121	79H	0111 1001
88	58H	0101 1000	122	7AH	0111 1010
89	59H	0101 1001	123	7BH	0111 1011
90	5AH	0101 1010	124	7CH	0111 1100
91	5BH	0101 1011	125	7DH	0111 1101
92	5CH	0101 1100	126	7EH	0111 1110
93	5DH	0101 1101	127	7FH	0111 1111
94	5EH	0101 1110	128	80H	1000 0000
95	5FH	0101 1111	129	81H	1000 0001

10進	16進	2進	10進	16進	2進
130	82H	1000 0010	164	A4H	1010 0100
131	83H	1000 0011	165	A5H	1010 0101
132	84H	1000 0100	166	A6H	1010 0110
133	85H	1000 0101	167	A7H	1010 0111
134	86H	1000 0110	168	A8H	1010 1000
135	87H	1000 0111	169	A9H	1010 1001
136	88H	1000 1000	170	AAH	1010 1010
137	89H	1000 1001	171	ABH	1010 1011
138	8AH	1000 1010	172	ACH	1010 1100
139	8BH	1000 1011	173	ADH	1010 1101
140	8CH	1000 1100	174	AEH	1010 1110
141	8DH	1000 1101	175	AFH	1010 1111
142	8EH	1000 1110	176	B0H	1011 0000
143	8FH	1000 1111	177	B1H	1011 0001
144	90H	1001 0000	178	B2H	1011 0010
145	91H	1001 0001	179	B3H	1011 0011
146	92H	1001 0010	180	B4H	1011 0100
147	93H	1001 0011	181	B5H	1011 0101
148	94H	1001 0100	182	B6H	1011 0110
149	95H	1001 0101	183	B7H	1011 0111
150	96H	1001 0110	184	B8H	1011 1000
151	97H	1001 0111	185	B9H	1011 1001
152	98H	1001 1000	186	BAH	1011 1010
153	99H	1001 1001	187	BBH	1011 1011
154	9AH	1001 1010	188	BCH	1011 1100
155	9BH	1001 1011	189	BDH	1011 1101
156	9CH	1001 1100	190	BEH	1011 1110
157	9DH	1001 1101	191	BFH	1011 1111
158	9EH	1001 1110	192	C0H	1100 0000
159	9FH	1001 1111	193	C1H	1100 0001
160	A0H	1010 0000	194	C2H	1100 0010
161	A1H	1010 0001	195	C3H	1100 0011
162	A2H	1010 0010	196	C4H	1100 0100
163	A3H	1010 0011	197	C5H	1100 0101

10進	16進	2 進	10進	16進	2 進
198	C6H	1100 0110	227	E3H	1110 0011
199	C7H	1100 0111	228	E4H	1110 0100
200	C8H	1100 1000	229	E5H	1110 0101
201	C9H	1100 1001	230	E6H	1110 0110
202	CAH	1100 1010	231	E7H	1110 0111
203	CBH	1100 1011	232	E8H	1110 1000
204	CCH	1100 1100	233	E9H	1110 1001
205	CDH	1100 1101	234	EAH	1110 1010
206	CEH	1100 1110	235	EBH	1110 1011
207	CFH	1100 1111	236	ECH	1110 1100
208	D0H	1101 0000	237	EDH	1110 1101
209	D1H	1101 0001	238	EEH	1110 1110
210	D2H	1101 0010	239	EFH	1110 1111
211	D3H	1101 0011	240	F0H	1111 0000
212	D4H	1101 0100	241	F1H	1111 0001
213	D5H	1101 0101	242	F2H	1111 0010
214	D6H	1101 0110	243	F3H	1111 0011
215	D7H	1101 0111	244	F4H	1111 0100
216	D8H	1101 1000	245	F5H	1111 0101
217	D9H	1101 1001	246	F6H	1111 0110
218	DAH	1101 1010	247	F7H	1111 0111
219	DBH	1101 1011	248	F8H	1111 1000
220	DCH	1101 1100	249	F9H	1111 1001
221	DDH	1101 1101	250	FAH	1111 1010
222	DEH	1101 1110	251	FBH	1111 1011
223	DFH	1101 1111	252	FCH	1111 1100
224	E0H	1110 0000	253	FDH	1111 1101
225	E1H	1110 0001	254	FEH	1111 1110
226	E2H	1110 0010	255	FFH	1111 1111

●コマンド索引●

◉ CONFIG.SYS コマンド

◉バッチコマンド

◉ MS-DOS6.2付属デバイスドライバ

◉ MS-DOS6.2 常駐プログラム

◉ MS-DOS6.2 コマンド

◉ ATOK8

◉ WX III

◉ ASPI マネージャ

◉ Melware

◉ Memory Server II

◉ PC カード用ドライバ/コマンド

◉その他のコマンド／ドライバ

●用語索引●

記号・数字・欧文

◉記号・数字・A〜C

◉ D〜L

和文

◉か行

◉さ行

◉た行

◉な行

◉は行

◉ま行

◉や・ら行

◉わ行

■著者略歴
寺口　俊伸（てらぐち・としのぶ）
1960年東京生まれ。
フリーテクニカルライターを経て、現在（有）フリーディア代表取締役。

■主な著書

入門エコロジーII（共著）	とことんこだわるCONFIG.SYS　改訂増補
入門ノートンユーティリティーズ	とことんこだわるファイル管理
入門MEMORY-PRO386	とことんこだわるNIFTY-Serve
入門MIFES Ver.5	バージョン4ユーザーのための一太郎Ver.5
入門Vzエディタ　マクロ編	とことんこだわるNIFTY-Serve大改訂
入門MIFES Ver.5.0　マクロ編	とことんこだわるCONFIG.SYS DOS/V編
入門MS-DOS Ver.5	（以上、エーアイ出版）

■カバーデザイン——藤田理一郎
■企　画・編　集——エーアイ出版企画編集部

とことんこだわるCONFIG.SYS NEC版MS-DOS6.2対応
（ディスク付）

1995年8月14日　第1版第1刷発行

著　書　寺口俊伸

発行者　村上　誠
発行所　エーアイ出版株式会社
〒150　東京都渋谷区桜丘町15－14　フジビル40
営　業　部　TEL：03(5458)5800　FAX：03(5458)5580
編　集　部　TEL：03(5458)6400　FAX：03(5458)6564
郵便振替　00140-9-175017

印　刷・製　本　バンスター株式会社

ISBN4-87193-388-1　　Printed in Japan

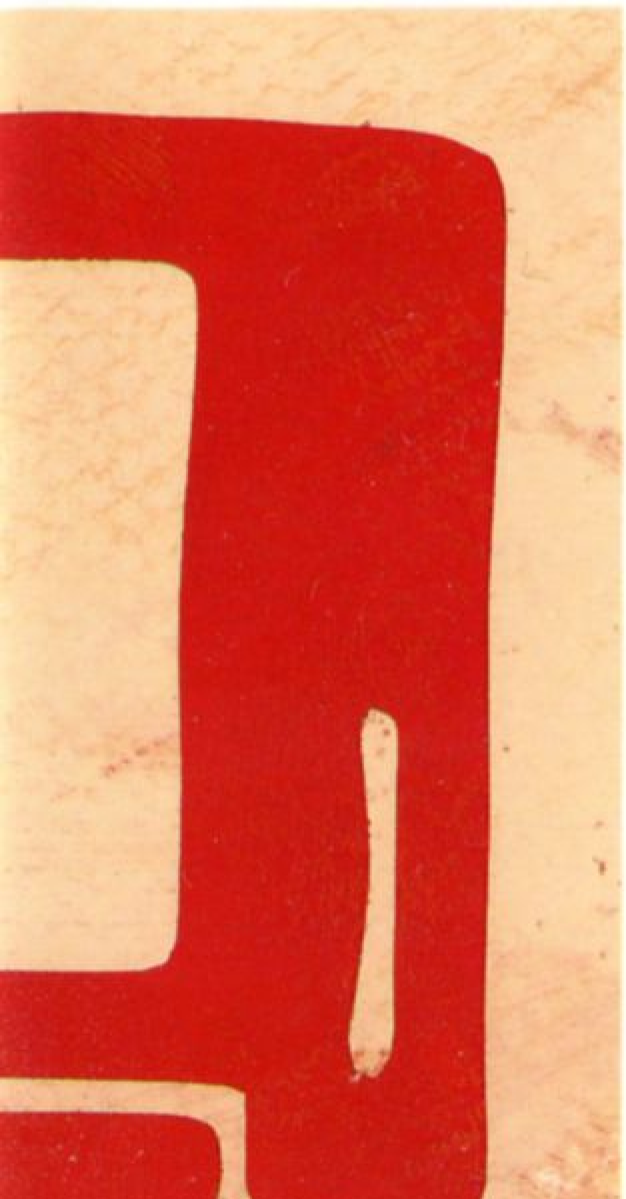
ISBN4-87193-388-1

C3055 P2000E

定価2000円(本体価格1942円)

9784871933889

1913055020000



PC98
シリーズ
対応版

とことんこだわる

CONFIG.SYS

NEC版MS-DOS6.2対応

エーアイ出版

PC98
シリーズ
対応版

とことん
こだわる

CONFIG.SYS

NEC版
MS-DOS
6.2対応

寺口俊伸著

エーアイ出版

＊388